

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Damjan Zajc

VPLIV USMERJANJA IN FILTRIRANJA PROMETA NA  
HITROST PRENOSA V KRAJEVNIH  
OMREŽJIH

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Ljubljana, 2009

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Damjan Zajc

VPLIV USMERJANJA IN FILTRIRANJA PROMETA NA  
HITROST PRENOSA V KRAJEVNIH  
OMREŽJIH

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Mojca Ciglarič

Ljubljana, 2009

Univerza  
v Ljubljani

Fakulteta za računalništvo  
in informatiko

Tržaška 25  
1000 Ljubljana, Slovenija  
telefon: 01 476 84 11  
faks: 01 426 46 47  
www.fri.uni-lj.si  
e-mail: dekanat@fri.uni-lj.si



Št. naloge: 00440/2009

Datum: 05.04.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DAMJAN ZAJC**

Naslov: **VPLIV USMERJANJA IN FILTRIRANJA PROMETA NA HITROST  
PRENOSA V KRAJEVNIH OMREŽJIH**

**THE EFFECTS OF ROUTING AND TRAFFIC FILTERING ON THE  
TRANSFER SPEED IN LOCAL AREA NETWORKS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Preglejte dosegljiva orodja za merjenje hitrosti prenosa med dvema omrežnima napravama. Izberite primerna orodja in razvijte potrebno dodatno funkcionalnost, da boste lahko testirali hitrosti prenosa podatkov s protokoloma TCP in UDP. Možno naj bo tudi testiranje hitrosti usmerjanja in filtriranja omrežnega prometa v odvisnosti od kompleksnosti pravil za filtriranje. Izmerite hitrosti prenosa v različnih tipih lokalnih omrežij, rezultate meritev pa ustrezno komentirajte. Ovrednotite tudi primernost in uporabnost razvitega orodja.

Mentor:

*M. Ciglaric*

doc. dr. Mojca Ciglaric



Dekan:

*Fran Solina*

prof. dr. Franc Solina

# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani      Damjan Zajc,


z vpisno številko      63030021,

sem avtor diplomskega dela z naslovom:

VPLIV USMERJANJA IN FILTRIRANJA PROMETA NA HITROST PRENOSA V  
KRAJEVNIH OMREŽJIH

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 15.10.2009 Podpis avtorja: 

# **ZAHVALA**

Zahvaljujem se mentorici, doc. dr. Mojci Ciglarič, in asistentu, univ. dipl. ing. Andreju Krevlu, za vodenje pri pisanju diplomske naloge. Zahvaljujem se tudi ženi Mojci Zajc, hčerki Neži Zajc, prijatelju Domnu Piškurju, svakinji Barbari Martinjak Zajc in sorodnikom, ki so kakorkoli pripomogli k nastanku tega dela.

# KAZALO

<b>POVZETEK.....</b>	<b>1</b>
<b>ABSTRACT .....</b>	<b>2</b>
<b>1. UVOD .....</b>	<b>3</b>
<b>2. RAČUNALNIŠKA OMREŽJA .....</b>	<b>4</b>
2.1. POVEZAVNA IN FIZIČNA PLAST .....	5
2.1.1. <i>Ethernet</i> .....	6
2.1.2. <i>MTU</i> .....	7
2.2. PROTOKOL IP .....	7
2.2.1. <i>Opis in delovanje</i> .....	7
2.2.2. <i>Naslov IP</i> .....	7
2.2.2.1. <i>Maska omrežja</i> .....	8
2.2.2.2. <i>Naslov omrežja</i> .....	8
2.2.2.3. <i>Naslov broadcast</i> .....	8
2.2.2.4. <i>Naslov naprave</i> .....	9
2.2.2.5. <i>Privzeti prehod</i> .....	9
2.2.3. <i>Fragmentiranje</i> .....	9
2.2.4. <i>Usmerjanje</i> .....	10
2.2.5. <i>Struktura datagrama IP</i> .....	11
2.3. TRANSPORTNA PLAST .....	12
2.3.1. <i>TCP</i> .....	12
2.3.1.1. <i>Segment TCP</i> .....	12
2.3.1.2. <i>Vrata</i> .....	13
2.3.1.3. <i>Nadzor pretoka</i> .....	14
2.3.1.4. <i>Vzpostavitev in rušenje povezave</i> .....	14
2.3.2. <i>UDP</i> .....	15
2.4. APLIKACIJSKA PLAST .....	15
<b>3. OPIS PROGRAMA IN NJEGOVIH DELOV .....</b>	<b>16</b>
3.1. OPIS PROBLEMA, FUNKCIONALNIH ZAHTEV ZA PROGRAM.....	16
3.2. UPORABLJENE TEHNOLOGIJE IN ORODJA.....	16
3.2.1. <i>Operacijski sistem Linux</i> .....	16
3.2.2. <i>Iptables</i> .....	16
3.2.3. <i>Iperf</i> .....	18
3.2.4. <i>Expect</i> .....	18
3.2.5. <i>Gnuplot</i> .....	18
3.2.6. <i>Delovanje programa</i> .....	18
3.2.6.1. <i>Arhitektura programa</i> .....	19
3.2.6.1.1. <i>Del namenjen merjencem</i> .....	20
3.2.6.1.2. <i>Del namenjen nadzorniku</i> .....	21
3.2.6.2. <i>Nastavitve programa</i> .....	21
3.2.6.3. <i>Nastavitve merjencev</i> .....	23
3.2.6.4. <i>Prenašanje programov in datotek</i> .....	24
3.2.6.5. <i>Povezava in zagon skript</i> .....	24
3.2.6.6. <i>Načini testiranja in delovanje</i> .....	24
3.2.6.6.1. <i>Optimalno testiranje UDP</i> .....	24
3.2.6.6.2. <i>Optimalno testiranje TCP</i> .....	25

3.2.6.7.	Shranjevanje testnih datotek.....	26
3.2.6.8.	Risanje grafov.....	27
<b>4.</b>	<b>REZULTATI TESTIRANJA IN UGOTOVITVE .....</b>	<b>27</b>
4.1.	REZULTATI .....	28
4.1.1.	<i>Omrežje 100 Mbit/s</i> .....	28
4.1.2.	<i>Omrežje 1 Gbit/s</i> .....	35
4.1.3.	<i>Ugotovitve</i> .....	42
<b>5.</b>	<b>ZAKLJUČEK .....</b>	<b>43</b>
<b>6.</b>	<b>LITERATURA .....</b>	<b>44</b>
	<b>SEZNAM SLIK, TABEL IN IZPISOV .....</b>	<b>46</b>

# SEZNAM UPORABLJENIH KRATIC

TCP	Transmission Control Protocol: <i>protokol za nadzor prenosa</i>
UDP	User Datagram Protocol: <i>nepovezovalni protokol</i>
IP	Internet Protocol: <i>internetni protokol</i>
MTU	Maximum Transmission Unit: <i>največja velikost okvirja</i>
ISO OSI	Open System Interconnection: <i>referenčni model, ki je sestavljen iz sedmih plasti, na vsaki plasti so določene posamezne omrežne funkcije</i>
CSMA/CD	Carrier Sense Multiple Access with Collision Detection: <i>kolizijski protokol z zaznavanjem trkov</i>
NIC	Network Interface Controller: <i>omrežni vmesnik</i>
CIDR	Classless Inter-Domain Routing: <i>brezrazredno notranje usmerjanje (za krajši zapis maske omrežja)</i>
RFCxxxx	Request for Comments: <i>dokumenti, v katerih je določenih večino standardov Internet</i>
ICANN	Internet Corporation for Assigned Names and Numbers: <i>organizacija, ki dodeljuje naslovni prosto IP</i>
GPL	General Public License: <i>licenca za odprto kodne programe</i>
NLANR/DAST	National Laboratory for Applied Network Research, Distributed Applications Support: <i>laboratorij v katerem so razvili program Iperf</i>
RJ45	Registered Jack 45: <i>tip priključka in vmesnik, ki ga uporabljamo za povezovanje v novjših Ethernet omrežjih, če je prenosni medij zvita parica</i>
BNC	Bayonet Neill-Concelman: <i>tip priključka, ki ga uporabljamo za povezovanje koaksialnega kabla na različne naprave</i>
TCP/IP	Transmission Control Protocol and Internet Protocol: <i>je model, ki je sestavljen iz petih plasti in je podoben modelu ISO OSI</i>
ADSL	Asymmetric Digital Subscriber Line: <i>tehnologija, ki omogoča hitrejši prenos podatkov preko parice</i>
TTL	Time To Live: <i>omejitev veljavnosti datagrama</i>



## **Povzetek**

Diplomsko delo vsebuje teoretični in praktični del. Teoretični del opisuje računalniška omrežja; kaj le-ta potrebujejo za svoje nemoteno delovanje, kako se med seboj ločijo in kako si med seboj izmenjujejo podatke.

Predstavljena sta protokola TCP in UDP, za kaj je vsak od njiju uporaben. Opisano je usmerjanje prometa, MTU, fragmentacija, idr.

V praktičnem delu je predstavljen program za testiranje hitrosti prenosa podatkov po omrežjih. Opisano je njegovo delovanje, komponente, ki jih potrebuje za nemoteno delovanje, nastavitve in kakšne rezultate dobimo ob končanem testiranju. Program testira hitrost omrežja z uporabo protokolov TCP in UDP v odvisnosti od števila pravil v požarnem zidu in velikosti okvirja Ethernet. Na koncu so razloženi rezultati testov in predstavljeni v obliki grafov, ki jih je izrisal program.

## **Ključne besede**

Datagram, protokol, Ethernet, MTU, fragmentacija, pasovna širina, testiranje hitrosti, požarni zid.

## **Abstract**

This thesis contains a theoretical and a practical section. The theoretical section describes computer networks. It describes their requirements for normal operation, the differences between different network types, and how data is exchanged among networked devices.

We present protocols TCP and UDP, and their fields of use. This section also describes traffic routing, MTU, fragmentation principles, etc.

The practical section contains a presentation of a network bandwidth measurement program. We present its operation, operational requirements, settings required and results produced by testing. Our program tests network speed (bandwidth) by using TCP and UDP protocols in relation to the number of rules applied to the firewall, and the size of the Ethernet frame. In the end, test results are explained, and presented in the form graphs rendered by the program.

## **Key words**

Datagram, protocol, MTU, fragmentation, bandwidth, network speed test, firewall.

# 1. UVOD

Napisal sem program za testiranje hitrosti omrežja z namenom, da bi ugotovil, kakšna je hitrost prenosa podatkov po omrežju oz. pasovna širina omrežja. Ta za osnovno delovanje uporablja program Iperf, ki omogoča testiranje s protokoloma TCP in UDP.

Program za delovanje potrebuje operacijski sistem Linux. Napisan je v skriptnem jeziku lupine Bash. Program s pomočjo drugih programov in ukazov lupine tvori celoto, ki omogoča samodejno testiranje hitrosti omrežja. Hitrost se testira na podlagi števila pravil v požarnem zidu, velikosti okvirja Ethernet (MTU) in velikosti sprejemnega okna v testih TCP. V diplomskem delu sem se predvsem osredotočil na število pravil v požarnem zidu. Zanimalo me je, kako vplivajo pravila na hitrost prenosa podatkov po omrežju.

Program omogoča testiranje tudi manjših usmerjevalnikov. Le-ti za testiranje potrebujejo programa Ssh, Iptables in lupino Bash.

Na podlagi testnih rezultatov lahko program izriše grafe.

Diplomsko delo vsebuje dve ključni poglavji:

- poglavje tri opisuje računalniška omrežja. V tem poglavju sem se osredotočil na naslavljanje naprav, usmerjanje prometa po omrežju, omejitve velikosti okvirja, sestavi okvirja Ethernet, sestavi datagrama TCP in UDP, sestavi segmenta IP in fragmentiranju,
- poglavje štiri vsebuje opis problema, ki sem ga rešil s pomočjo programa, uporabljene tehnologije in orodja pri sami izdelavi programa, opis delovanja programa, shranjevanje testnih podatkov in izdelava grafov.

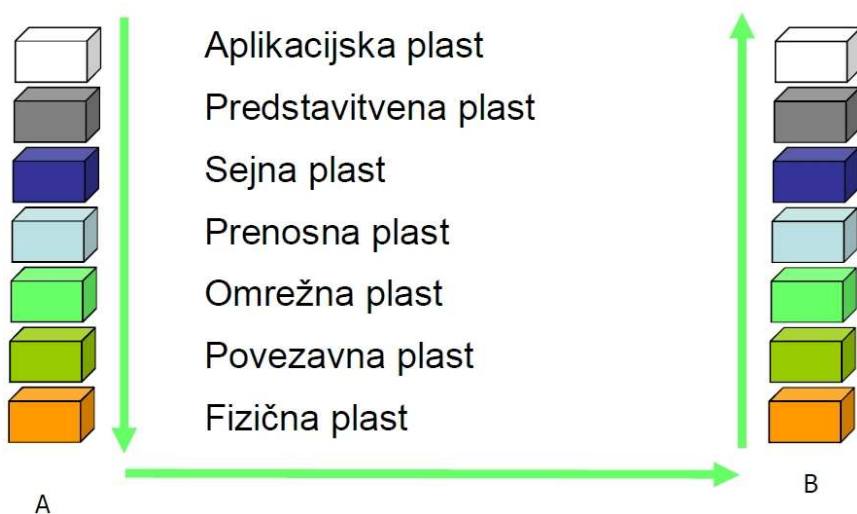
## 2. RAČUNALNIŠKA OMREŽJA

Referenčni model ISO OSI je sestavljen iz sedmih plasti. Na vsaki plasti so določene posamezne omrežne funkcije. Model je bil razvit leta 1984 in velja za osnovni arhitekturni model komunikacije med računalniki. Model ISO OSI je sistematičen in konceptualno zasnovan, vendar je le referenčni model, ki v celoti ni nikoli zaživel.

Plasti modela ISO OSI:

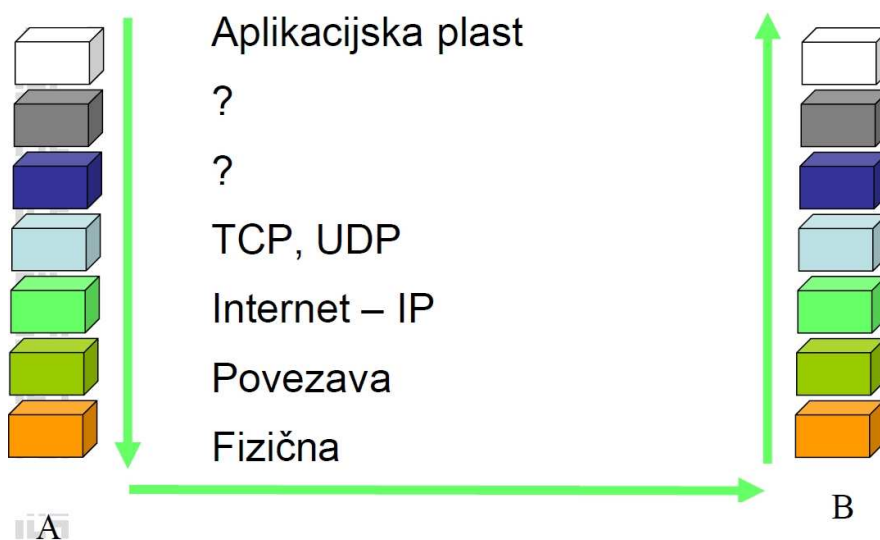
- aplikacijska plast: vmesnik med uporabnikom in modelom ISO OSI,
- predstavitevna plast: zagotavlja različne načine kodiranja in sisteme pretvorb za aplikacijsko plast,
- sejna plast: nadzira komunikacijo med računalniki,
- transportna plast: plast definira način prenosa, dolga sporočila razbije na manjše dele, odkriva in odpravlja napake,
- omrežna plast: preklapljanje (povezavne in nepovezavne storitve), usmerjanje in izogibanje zamašitvam,
- povezavna plast: način komunikacije (asinhrono ali sinhrono), zaznavanje in odpravljanje napak, kontrola pretoka,
- fizična plast: predpisuje prenosni medij preko katerega se prenašajo podatki, definira nivo signala, hitrost prenosa, način zapisa podatkov.

Entitetni par je par procesov, ki komunicirata na isti plasti (na različnih straneh). [30]



Slika 1. Sestavni deli modela ISO OSI. [29]

Model TCP/IP je bil razvit v sedemdesetih letih iz tehnologij uporabljenih v ARPANET-u. Za razliko od modela ISO OSI, je bolj prilagodljiv in fleksibilen, kar je tudi razlog za njegovo vse večjo uporabo. Model TCP/IP je dejansko pet plasten model, ki ima sejno in predstavitevno plast združeno v aplikacijsko plast, fizična in povezavna plast pa sta pogosto združeni v plast omrežje. Naloge posameznih plasti so opisane pri modelu ISO OSI, podrobneje pa so predstavljene v nadaljevanju. [3]



Slika 2. Sestavni deli modela TCP/IP. [29]

## 2.1. Povezavna in fizična plast

Fizična plast je prva po hierarhiji, ki jo določa model TCP/IP (Slika 2). Naloge fizične plasti so:

- prenos bitov po prenosnem kanalu: v analogni ali digitalni obliki,
- kodiranje: način zapisa bita,
- pretvorba električnih signalov v obliko za prenos po mediju,
- multipleksiranje: razdeljevanje prenosnega kanala.[15]

Naprava lahko preko prenosnega kanala prenese paket (okvir) po prenosnem mediju. Prenosni kanal je lahko:

- enosmerni ali dvosmerni,
- serijski ali paralelni (zaporedni ali vzporedni),
- dvo-točkovni ali skupinski.

Fizična plast predpisuje (signal, hitrost idr.) prenosni medij, preko katerega se prenašajo podatki. Primeri prenosnih medijev so:

- fizični prenos pomnilniških medijev,
- parica in zvita parica,
- koaksialni kabel,
- optično vlakno,
- brezžične povezave. [1, 9]

Povezavna plast je druga po hierarhiji, ki jo določa model TCP/IP (Slika 2). Njene naloge so:

- prenos okvirja po povezavi iz enega vozlišča do drugega vozlišča,
- dostop do skupnega medija,
- naslavljanje,
- odkrivanje napak ter kontrola pretoka med sosednjimi vozlišči. [2, 3, 9]

Fizično in povezavno plast v modelu TCP/IP imenujemo »omrežje«.

### 2.1.1. Ethernet

Preambula	Ponorni naslov	Izvorni naslov	vrsta	podatki	CRC
-----------	-------------------	-------------------	-------	---------	-----

Slika 3. Vsebina okvirja Ethernet. [9]

Okvir Ethernet vsebuje (Slika 3):

- preambula: služi za sinhronizacijo uri oddajnika in prejemnika,
- ponorni naslov: če se ujema z naslovom prejemnika ali če je naslov broadcast, okvir posreduje naprej omrežni plasti, sicer ga zavrže,
- izvorni naslov: je naslov tistega, ki pošilja okvir,
- vrsta: omrežni protokol (IP, ARP idr.),
- podatki: datagram dolžine 46 – 1500 B,
- CRC: če ni pravilen, se okvir zavrže (kontrolni biti).

Ethernet je prvotno deloval s hitrostjo 10 Mbit/s. Za dostop do skupinskega medija uporablja protokol z zaznavanjem prenosa in odkrivanjem trkov (CSMA/CD). CSMA/CD ima naslednje lastnosti:

- zvezni čas,
- posluša pred oddajo,
- v primeru kolizije preneha z oddajo,
- pred ponovno oddajo čaka naključno časovno obdobje,
- v primeru več zaporednih kolizij vsakič dlje čaka.

Trenutne nadgradnje tega protokola delujejo s hitrostjo 100 Mbit/s (Fast Ethernet), 1 Gbit/s (Gigabit Ethernet) in 10 Gbit/s. [9]

### 2.1.2. MTU

Maximum transmission unit (MTU) je vrednost, ki določa največjo velikost okvirja Ethernet v določenem omrežju. Vrednost, ki je lahko med 68 B in 64 KB, je odvisna od zmogljivosti fizične plasti (npr. v Ethernet 100 Mbit/s je največja velikost 1500 B). Tabela 1 prikazuje MTU v različnih tipih omrežij.

Tip omrežja	MTU	Hitrost
Ethernet	1500 B	10 Mbit/s – 1 Gbit/s
Ethernet - Jumbo Frames	9000 B	1 Gbit/s – 10 Gbit/s
802.11 (wifi – brezžično omrežje)	2272 B	54 Mbit/s – 300 Mbit/s
PPPoE – ADSL (IEEE 802.3/802.2)	1492 B	512 Kbit/s – 8 Mbit/s
Dial-up (X.25)	576 B	56 Kbit/s – 114Kbit/s

Tabela 1. Seznam omrežij, njihove velikosti okvirjev (MTU) in teoretične hitrosti.

## 2.2. Protokol IP

### 2.2.1. Opis in delovanje

Protokol IP zna povezati več različnih podomrežij. Njegova naloga je kar najhitreje zagotoviti prenos datagrama od izvora do ponora, ne glede na to, če so vmes druga omrežja.

Teoretično je možno, da je datagram IP velik do 64 KB. V praksi ponavadi velikost ne presega 1500 B, zato da ustreza zmogljivosti fizične plasti.

Datagram se lahko pošlje preko usmerjevalnikov v druga omrežja, se po potrebi fragmentira in je ponovno sestavi na strani prejemnika. Več o fragmentiranju piše v točki 2.2.3. [3]

### 2.2.2. Naslov IP

Naslov IP je 32-bitna beseda, ki jo zapišemo desetiško kot xxx.xxx.xxx.xxx, kjer predstavlja xxx skupino števil od 0 – 255. V desetiškem zapisu so skupine, zaradi preglednosti, med seboj ločene s piko. Če naslov IP napišemo v binarni obliki (xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx), je vrednost posameznega »x« 0 ali 1. Za računalnik je naslov IP dejansko 32-bitna beseda.

Naprava ima lahko več naslovov IP. Če je priključena na dve različni omrežji potrebuje dva naslova, kar pa ne pomeni, da mora imeti fizično dva ločena mrežna vmesnika (dve omrežni kartici – NIC) (Slika 4).



Slika 4. Omrežna kartica (NIC) s priklopoma BNC in RJ45. [27]

Poleg naslova IP poznamo še:

- masko omrežja,
- naslov omrežja,
- naslov broadcast,
- privzeti prehod (ang. default gateway),
- naslov naprave.

Vsi ti so ravno tako 32-bitne besede. V nadaljevanju bom opisal vsakega od njih.

### 2.2.2.1. Maska omrežja

Maska omrežja določa dolžino naslova omrežja ali podomrežja. Gre za 32-bitni niz, ki ima enice na mestih, ki označujejo naslov omrežja, na ostalih so ničle. Maska /24 (notacija CIDR) pomeni, da je prvih 24 bitov naslov omrežja, zadnjih 8 (branih z leve proti desni) pa naslov naprave. Na primer:

maska omrežja 255.255.255.0 ali /24 binarni zapis: 11111111 11111111 11111111 00000000

### 2.2.2.2. Naslov omrežja

Naslov omrežja je naslov IP, kjer so vsi biti, ki določajo naslov naprave enaki nič.

naslov naprave	192.168.0.10	binarni zap.: 11000000 10100100 00000000 00001010
maska omrežja	255.255.255.0	binarni zap.: 11111111 11111111 11111111 00000000
naslov omrežja	192.168.0.0	binarni zap.: 11000000 10100100 00000000 00000000

### 2.2.2.3. Naslov broadcast

Naslova broadcast nima nobena naprava v omrežju. Le-ta služi pošiljanju okvirja vsem napravam v omrežju. Torej naprava, ki pošlje poizvedbo na ta naslov, dejansko pošlje podatke vsem napravam na tem omrežju.

Naslov broadcast ima vse bite, ki določajo naslov naprave, postavljene na 1.

naslov naprave	192.168.0.10	binarni zap.: 11000000 10100100 00000000 00001010
maska omrežja	255.255.255.0	binarni zap.: 11111111 11111111 11111111 00000000
naslov broadcast	192.168.0.255	binarni zap.: 11000000 10100100 00000000 11111111



#### 2.2.2.4. Naslov naprave

Vzemimo primer omrežja z masko 255.255.255.0 (/24 – notacija CIDR). To omrežje ima 256 različnih naslovov IP. Dva sta namenjena naslovu broadcasta in naslovu omrežja, ostalih 254 pa naslovom naprav. Dejansko imamo lahko v vsakem omrežju  $2^n - 2$  naprav (kjer je  $n$  število bitov, ki so namenjeni naslovom IP v omrežju).

maska omrežja	255.255.255.0
naslov omrežja	192.168.0.0
naslov broadcast	192.168.0.255
naslovi naprav	od 192.168.0.1 do 192.168.0.254

#### 2.2.2.5. Privzeti prehod

Privzeti prehod (ang. default gateway) potrebujemo takrat, kadar želimo promet, ki ni namenjen našemu omrežju, poslati drugemu omrežju. Gre za naslov naprave (usmerjevalnika), ki je povezana z našim omrežjem in še z najmanj enim. Več o usmerjanju (rabi privzetega prehoda) piše v točki 2.2.4.

### 2.2.3. Fragmentiranje

Vsako omrežje ima omejitve. Ena od omejitev je velikost okvirja na fizični plasti (MTU), ki je od omrežja do omrežja lahko različna. Vzroki za to so:

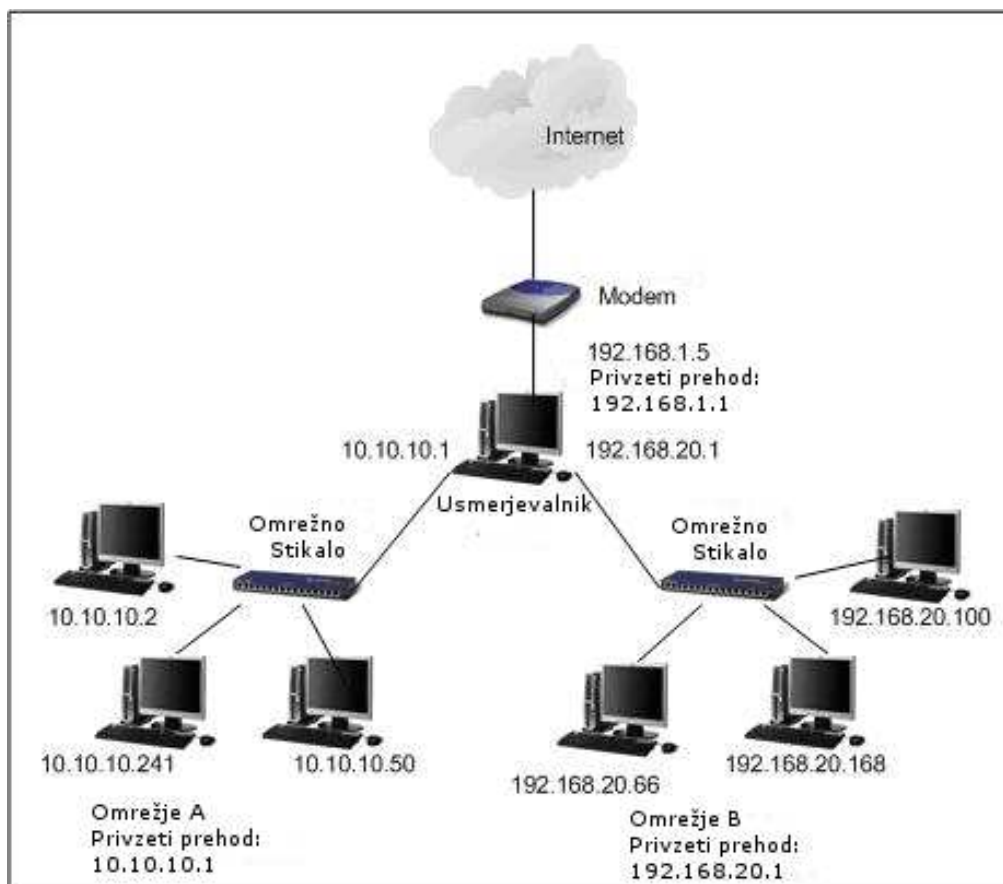
- strojne omejitve,
- omejitve operacijskega sistema (npr. vsi predpomnilniki so 512 B),
- omejitve protokola (št. bitov v polju za dolžino paketa),
- nacionalni ali mednarodni standardi,
- želja po zmanjšanju ponovnega oddajanja zaradi napak.

Fragmentacijo lahko izvaja odjemalec ali usmerjevalnik, ki dobi prevelik datagram IP. Letega fragmentira na njemu ustrezne manjše fragmente. Posamezni paket se lahko fragmentira večkrat, vsakič ko paket naleti na omejitev (MTU). Fragmenti nato potujejo do prejemnika, ki jih sestavi nazaj v celoto.

Slabost fragmentacije je v tem, da se lahko izvaja večkrat in zato lahko naraste obremenitev omrežja. Zaradi fragmentacije lahko pride do večje obremenitve, saj:

- so glave lahko večje od pripetih podatkov,
- se v primeru izgube enega samega fragmenta, postopek ponovi (prejemnik zahteva ponovno pošiljanje).

## 2.2.4. Usmerjanje



Slika 5. Usmerjevalnik in tri (pod)omrežja. [28]

Če želimo prenesti promet iz enega omrežja v drugo, potrebujemo usmerjevalnik. Le-ta usmerja promet med omrežji, s katerimi je povezan. Na vseh omrežjih morajo imeti naprave nastavljen privzeti prehod tako, da kaže na usmerjevalnik (Slika 5).

### 2.2.5. Struktura datagrama IP

4	4	8	16	
Različica	IHL	Tip storitve	Skupna dolžina	
Identifikacija			Zastavice	Zamik fragmenta
TTL	Protokol		Kontrolna vsota - glava	
Izvorni IP naslov				
Ponorni IP naslov				
Možnosti				Zapolnjevanje
Podatki				

Slika 6. Sestavni deli datagrama IP. [22]

Glava datagrama IP vsebuje (Slika 6):

- različica IP: različica uporabljenega protokola IP (najpogosteje IPv4 ali IPv6),
- IHL: velikost glave datagrama IP v bajtih,
- tip storitve: način prenosa oz. na kakšen način želim prenesti datagram,
- skupna dolžina: dolžina datagrama skupaj z glavo,
- identifikacija: za fragmentacijo in sestavljanje,
- zastavice: za fragmentacijo in sestavljanje,
- zamik fragmenta: za fragmentacijo in sestavljanje,
- TTL: omejitev veljavnosti datagrama, pove koliko skokov naredi paket preden se smatra za izgubljenega (največja vrednost je 255),
- protokol: protokol transportne plasti - ponavadi UDP ali TCP ali drugi,
- kontrolna vsota glave: preverja glavo in se uporablja za zaznavanje napak,
- izvorni naslov IP: naslov IP pošiljatelja,
- ponorni naslov IP: naslov IP prejemnika,
- opcije: ponavadi jih ni, če pa so: varnost, časovna oznaka, začrtana pot, posnemi pot ipd.

## 2.3. *Transportna plast*

Transportna ali prenosna plast se nahaja med aplikacijsko in omrežno plastjo (Slika 2). Njena naloga je prenos podatkov med dvema končnima napravama. Sprejema zahteve aplikacijske plasti in jih izvršuje s pomočjo omrežne plasti. Transportna plast je tako vmesnik med informacijskim sistemom in omrežjem. Aplikacijski plasti zagotavlja storitve popolnega transportnega kanala. Poleg naloge prenosa podatkov ima še nalogo odpravljanja napak, kontrolo pretoka in drugo. Med najbolj znane implementacije transportne plasti spadata protokola TCP in UDP.[4]

### 2.3.1. TCP

TCP je formalno opisan v dokumentih RFC793 [5] , RFC1122 [16] , RFC1323 [17] , RFC2018 [18] , RFC2581 [19] . Razvit je bil posebej za zanesljiv prenos podatkov po nezanesljivih omrežjih. Se dinamično prilagaja lastnostim omrežja in se robustno odziva na različne napake.

Transportna plast prejema pritoke podatkov iz lokalnih procesov, ki zahtevajo prenos podatkov po omrežju. Podatke razbije v segmente, ki ne presegajo velikosti 64 KB (v praksi je velikost ponavadi 1460 podatkovnih bajtov, zato da se količina podatkov prilega v širino okvirja Ethernet (2.1.2) skupaj s podatki in glavo TCP in IP). Tako lahko omrežna plast pošlje vsak segment transportne plasti kot ločen paket. Ko paketi, ki vsebujejo segment TCP dospejo do druge naprave, so predani prejemnikovi transportni plasti, ki ponovno sestavi prvotni tok podatkov. Plast IP ne zagotavlja pravilne dostave paketov, zato jih TCP po potrebi ponovno pošlje (ko se izteče čas (ang. time out) ali dobi zahtevo za ponovno pošiljanje). Paketi lahko dospejo v napačnem vrstnem redu, tako je naloga TCP-ja tudi ta, da jih pravilno sestavi. [3]

#### 2.3.1.1. Segment TCP

0-3	4-7	8-15	16-31
Izvorna vrata			Ponorna vrata
Zaporedna številka			
Številka potrditve			
Odmik	Rezervirano	Zastavice	Okno
Kontrolna vsota		"Nujni" kazalec	
Možnosti			
Podatki			

Slika 7. Sestavni deli segmenta TCP. [22]

Velikost glave segmenta TCP je 20 bajtov. Ta je sestavljena iz:

- izvornih in ponornih vrat: vsaka po 4 bajte,
- zaporedne številke segmenta: zagotovi končno urejenost po vrstnem redu,
- številke potrditve: zaporedna številka segmenta – uporablja se v kombinaciji z zastavico ACK,
- odmika: podatek o dolžini glave,
- rezerviranega prostora: neuporabljen prostor (6 bitov),
- zastavice: URG – dvigne prioriteto paketa, ACK – za potrditve paketa, PSH – zahteva takojšnje posredovanje paketa, da ne čaka v pomnilniku, RST – sproži ponovno vzpostavitev povezave ali pa zavrne poskus odpiranja povezave, SYN – se uporablja za vzpostavitev povezave, FIN – se uporablja za rušenje povezave,
- velikosti okna: vpis praznega prostora v sprejemnem vmesniku,
- kontrolne vsote: »checksum« služi za preverjanje pristnosti podatkov,
- »nujnega« kazalca: informacija, kje se nahajajo nujni podatki – uporablja se v kombinaciji z zastavico URG,
- možnosti: prostor za dodatne možnosti, ki niso zajete v glavi segmenta,
- podatkov: prostor namenjen podatkom.

### 2.3.1.2. Vrata

Protokol TCP potrebuje vrata, da preda podatke pravemu procesu. Le-ta imajo lahko vrednost od 0 do 65535. [6]

Seznam vrat:

- od 0 do 1023: vrednosti vrat, ki jih uporabljajo znane aplikacije (ang. well-known ports),
- od 1024 do 49151: vrednosti vrat, ki so registrirana pri organizaciji ICANN. (ang. registered ports), [21]
- od 49152 do 65535: vrednosti vrat, ki jih sistem sam dodeljuje (ang. dynamic ports).

VRATA	PROTOKOL	OPIS
20, 21	TCP	FTP
22	TCP	SSH, SFTP, SCP
25	TCP	SMTP
53	TCP, UDP	DNS
80	TCP	HTTP

Tabela 2. Seznam vrat, ki jih uporabljajo nekateri protokoli.

### 2.3.1.3. Nadzor pretoka

V primeru obremenitve, ki je večja kot jo omrežje lahko obvlada, se pojavi zasičenje omrežja. Internet pri tem ni nobena izjema. Rešitev le-tega v večini opravi sam protokol TCP, saj omogoča upočasnitev pošiljanja podatkov. TCP skuša to doseči tako, da dinamično spreminja velikost sprejemnega in zamašitvenega okna (ang. window size and congestion window).

Vzroka za zakasnitve sta lahko dva:

- sprejemnik ne uspe dovolj hitro obdelati prometa,
- napake v omrežju (okvare, preobremenitev ipd.).

Da pošiljatelj ugotovi vrsto težave in njeno rešitev, vzdržuje dve okni:

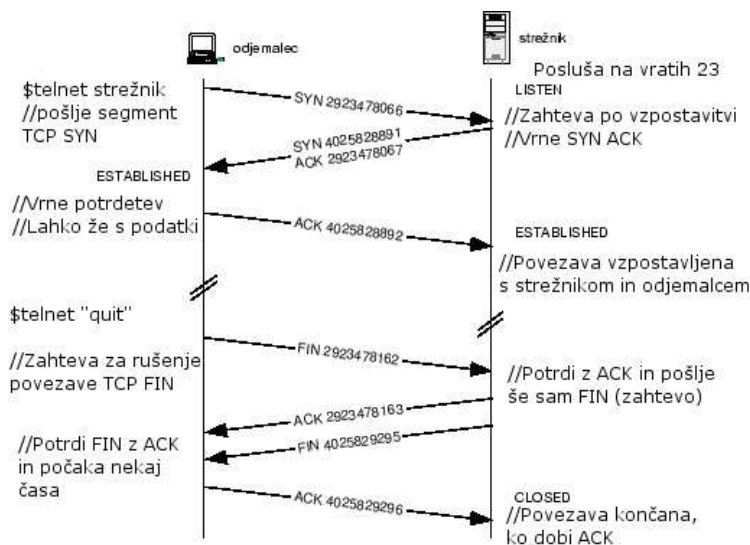
- sprejemno okno, ki ga je odobril prejemnik (ang. receive window),
- zamašitveno okno, ki ga vzdržuje sam (ang. congestion window).

Pošiljatelj pošlje število bitov, ki ustreza najmanjši vrednosti od obeh oken. [3, 9]

### 2.3.1.4. Vzpostavitev in rušenje povezave

Mehanizem vzpostavljanja in rušenja povezave TCP je prikazan na sliki 8. Sistem, ki zahteva vzpostavitev povezave, v našem primeru je to odjemalec na levi strani slike, pošlje strežniku paket podatkov z oznako SYN. Strežnik odgovori s paketom SYN, ki ima oznako ACK in z njim potrdi odjemalčevo zahtevo. Odjemalec potrdi sprejem tega odgovora z novim paketom z oznako ACK. S tem je povezava vzpostavljena. Tako povezovanje se imenuje tri smerno rokovanje (ang. three-way handshake).

Tudi rušenje povezave poteka po dogovorjenem postopku. Odjemalec zahteva prekinitev seje tako, da pošlje paket z oznako FIN. Strežnik to zopet potrdi z ACK in tudi sam pošlje paket z oznako FIN. Ko odjemalec potrdi sprejem tega paketa s svojim ACK, je s tem seja zaključena. [7]



Slika 8. Vzpostavljanje in rušenje povezave po protokolu TCP. [7]

### 2.3.2. UDP

Protokol UDP je opisan v dokumentu RFC768 [20]. Je transportni protokol, ki ne potrebuje vzpostavitve povezave. Pri protokolu UDP je segment sestavljen iz 8 bajtne glave, kateri sledijo podatki. Glava protokola UDP je prikazana na sliki 9.

Glava vsebuje podatke o izvornih in ponornih vratih ter dolžini (število bajtov segmenta UDP vključno z glavo) in kontrolni vsoti (ang. checksum) segmenta, ki pa je opcijška.

UDP ne regulira pretoka podatkov, nima kontrole zamašitev ter ponovnega oddajanja v primeru izgube ali okvare segmenta. Edina prednost dodatne uporabe UDP, pred direktno uporabo prenosa preko IP, je samo v dodani informaciji o izvornih in ponornih vratih, tako da aplikacijska plast ve, kateremu procesu segment pripada.

Pri protokolu UDP pogosto odjemalec pošlje strežniku zahtevo in pričakuje kratek odgovor. Če se zahteva ali odgovor izgubi, odjemalec samo počaka, da preteče predvideni čas za odgovor (time out) in ponovno pošlje zahtevo. Prednost tega protokola je, da potrebuje manj sporočil, saj protokol ne zahteva predhodnega rokovanja (predhodne vzpostavitve povezave (ang. three way handshake)).

Proces, ki uporablja protokol UDP na ta način, je DNS (Domain Name Server). Nek program lahko pošlje paket UDP, ki vsebuje ime gostitelja, strežniku DNS. Strežnik odgovori s paketom UDP, ki vsebuje gostiteljev naslov IP (npr. 212.235.188.25). Za takšno delovanje ne potrebuje vzpostavljanja ali rušenja povezave. Če torej ni izgub, gresta skozi omrežje samo dva paketa.

0-15	16-31
Izborna vrata	Ponorna vrata
Dolžina	Kontrolna vsota
Podatki	

Slika 9. Sestavni deli segmenta UDP. [22]

## 2.4. Aplikacijska plast

Med seboj komunicirajo procesi in ne programi. Program lahko dostopa do standardnih ali nestandardnih storitev. Brskalniki uporabljajo standardno storitev http. Omrežne aplikacije tečejo na končnih napravah. Odjemalec in strežnik predstavljata par procesov, ki si izmenjujeta sporočila. Aplikaciji ni potrebno poznati spodnjih plasti. Naslov naprave (ang. host address) in naslov procesa (številka vrat) potrebujemo, da naslovimo proces na drugi strani. Vtič (ang. socket) je vmesnik med aplikacijsko in transportno plastjo.[26]

### 3. OPIS PROGRAMA IN NJEGOVIH DELOV

#### 3.1. *Opis problema, funkcionalnih zahtev za program*

Naloga diplomskega dela je bila izmeriti hitrosti, ki jih dosegajo različni računalniki (bodisi namenski ali osebni) pri usmerjanju prometa. V ta namen sem napisal program, ki testira hitrost usmerjanja z različnimi parametri. Ti so:

- število pravil v požarnem zidu,
- velikost okvirja Ethernet,
- velikost sprejemnega okna pri testih TCP.

Funkcionalne zahteve programa:

- testiranje hitrosti pretoka s protokoloma TCP in UDP,
- dinamično spreminjanje sprejemnega okna pri testiranju s protokolom TCP,
- dinamično spreminjanje velikosti okvirja (MTU) pri testiranju s protokoloma TCP in UDP,
- spreminjanje parametrov (čas, način, število pravil) skupin testov,
- rezultati testne skupine se shranijo v datoteko z enoličnim imenom,
- testiranje deluje samodejno,
- shranjevanje rezultatov na zanesljiv medij (v primeru izpada električnega toka),
- izdelava grafov iz testnih rezultatov.

#### 3.2. *Uporabljene tehnologije in orodja*

##### 3.2.1. Operacijski sistem Linux

Linux je operacijski sistem podoben Unixu s prosto dostopno izvorno kodo, zaščiteno s splošnim dovoljenjem GNU (GNU General Public License (GPL)). Podpira večprocesorski način delovanja, številne datotečne sisteme, možnost izbire razvrščanja itd.

Obstaja več distribucij Linux, kot so Debian, RedHat, Fedora, Scientific Linux idr. Za diplomsko delo sem uporabil Scientific Linux različico live 52. Live pomeni, da za delovanje ni potrebno predhodno namestiti sistema na trdi disk. [8]

##### 3.2.2. Iptables

Iptables za svoje delovanje uporablja tri že vnaprej določene tabele:

- mangle,
- filter,
- nat.

Vsaka tabela, razen tabela filter, vsebuje naslednje verige:

- PREROUTING (paket bo vstopil v to verigo, še preden bo odločeno o njegovem usmerjanju)
- INPUT (paket je namenjen tej napravi)
- FORWARD (vsi paketi, ki so bili usmerjeni in niso za to napravo, bodo prečili to tabelo)

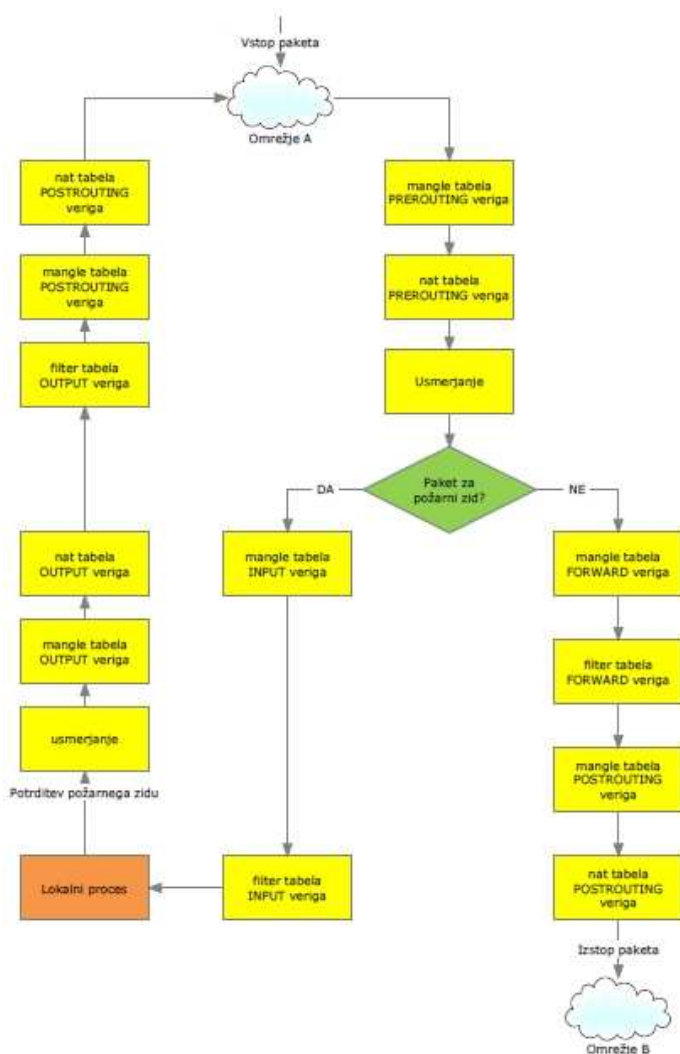


- OUTPUT (to tabelo bodo prečili tisti paketi, ki so bili poslani od same naprave)
- POSTROUTING (odločitev za usmerjanje je bila izvršena in paketi vstopijo v to tabelo, tik preden jih le-ta preda strojni opremi)

Tabela filter ne vsebuje PREROUTING in POSTROUTING verige. Slika 10 prikazuje graf poteka prečenja verig v jedru Linux-u.

Iptables je orodje, ki omogoči nastavljanje tabel tako, da operacijski sistem ve kako se odzvati na določene pakete. Različni moduli se trenutno uporabljajo za različne protokole. Iptables za IPv4, ip6tables za IPv6, arptables za ARP itd. Jaz sem uporabil modul iptables. Za zagon in konfiguracijo iptables se je potrebno prijaviti kot root uporabnik.

Vsaka tabela je povezana z različnim tipom obdelave paketov. Paketi so obdelani tako, da postopoma prečijo verigo pravil. Pravilo v določeni verigi lahko povzroči preskok v drugo verigo pravil in to se lahko ponavlja tako dolgo dokler je potrebno. Lastnosti paketa (naslov izvora, naslov ponora, uporabljen protokol, vrednost vrat TCP ali UDP in zastavice v glavi datagrama IP) določijo, katero tabelo bo prečil najprej. [10]



Slika 10. Diagram poteka – potovanja omrežnega paketa (paketa IP) po požarnem zidu v jedru Linux.

### 3.2.3. Iperf

Iperf je odprto kodno orodje napisano v programskem jeziku C++, namenjeno za testiranje omrežij. Iperf deluje kot običajna TCP ali UDP storitev - na eni strani povezave je strežnik, na drugi odjemalec. Strežnik je v vlogi prejemnika prometa, odjemalec pa tvori in pošilja promet. Iperf izmeri količino prenešenih podatkov in na podlagi znanega trajanja meritve izračuna hitrost prenosa podatkov. Pri tem moramo upoštevati, da Iperf izpiše hitrost glede na količino prenešenih podatkov brez glav TCP, IP ter okvirja Ethernet. V lokalnem 100 Mbit/s omrežju Ethernet lahko dosežemo prepustnost do približno 94 Mbit/s. [11] Orodje so izdelali v laboratoriju NLANR/DAST (ang. "National Laboratory for Applied Network Research, Distributed Applications Support"). Po zaključku projekta NLANR se razvoj Iperf nadaljuje na Sourceforge, kjer se nahaja tudi izvorna koda programa tako za Linux kot za operacijski sistem Windows. Program je prosto dostopen pod licenčnimi pogoji Univerze v Illinoisu. [11]

### 3.2.4. Expect

Je neke vrste skriptni jezik, ki deluje po načelu pričakovanj. Če na primer nek program, ki ga expect nadzira (posluša), zahteva geslo, expect iz svojega seznama pričakovanj vzame podatke za geslo in mu jih posreduje. Tipični programi, ki jih expect nadzira, so Ssh, Sftp, Scp idr. [12]

### 3.2.5. Gnuplot

Gnuplot je program, ki zna risati tako dvodimenzionalne, kot tudi tridimenzionalne grafe. Grafe lahko sproti prikazuje na zaslon ali pa izvozi v png, jpeg, eps, svg idr. oblike. Napisan je v programskem jeziku C. Z njim lahko upravljamo preko opravilne vrstice. Uporabljajo ga različni programi in programski jeziki. Deluje skorajda na vseh operacijskih sistemih kot so Microsoft Windows, Linux, Unix, Mac OS x idr. [13]

### 3.2.6. Delovanje programa

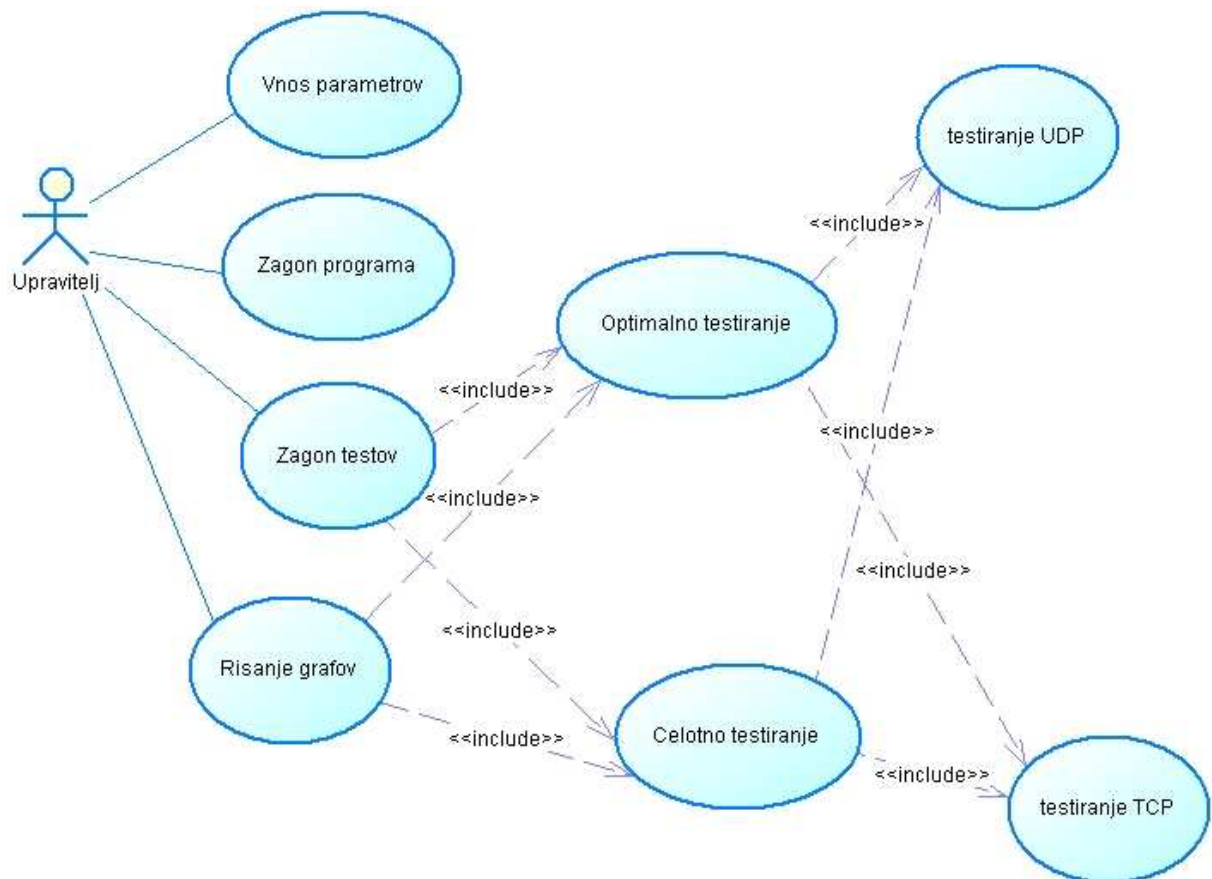
Program, ki sem ga izdelal, omogoča samodejno testiranje hitrosti omrežja. Za samodejno testiranje je potrebno, da se predhodno nastavi vse parametre za njegovo delovanje. Testiranje zajema štiri naprave. Ena od njih, ki se imenuje nadzornik, nadzoruje ostale. Ostale so merjenci, ki imajo naslednje nazive:

- odjemalec,
- strežnik,
- usmerjevalnik.

### 3.2.6.1. Arhitektura programa

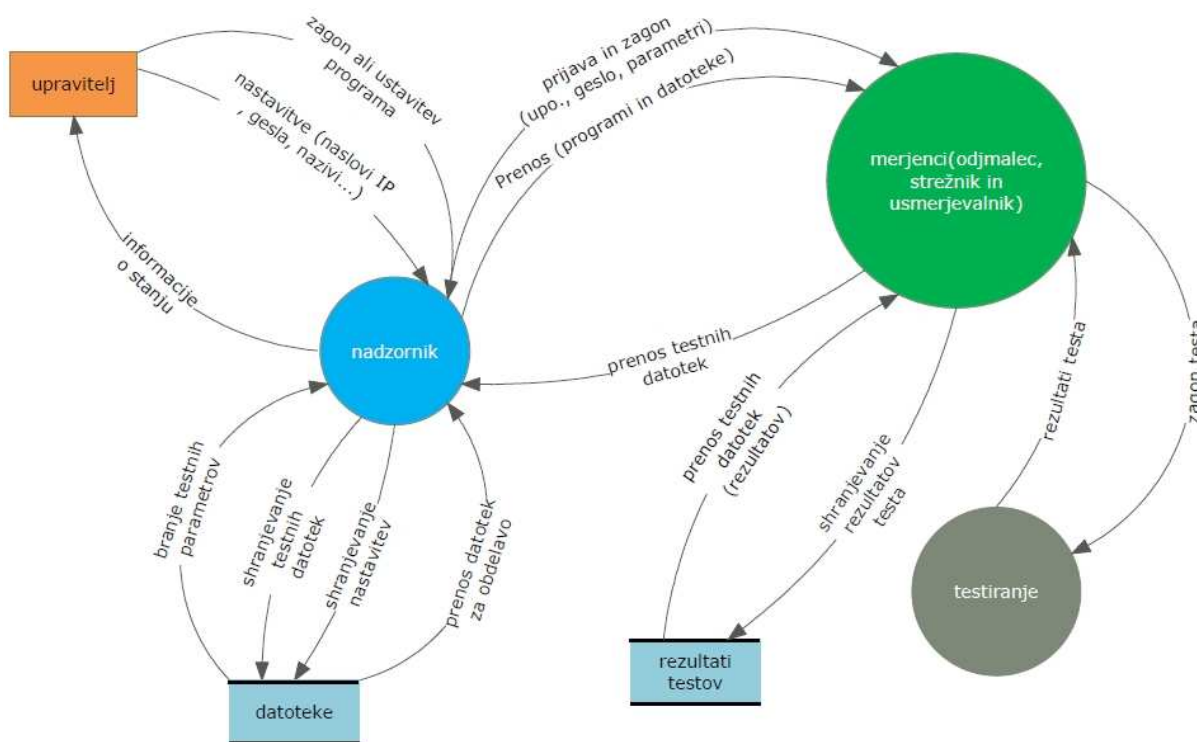
Diagram primerov uporabe na sliki 11 nam prikazuje različna rokovanja s testnim programom:

- vnos parametrov: v datoteko »settings« in »testFile« vnesemo vse potrebne parametre,
- zagon programa: program zaženemo z ukazom »./runProgram.sh« ali s pozicijskim argumentom,
- zagon testov: po zagonu programa se nam izpiše meni, iz katerega izberemo možnost »zaženi testiranje«,
- optimalno testiranje: vsebuje kratko testiranje meje protokola,
- celotno testiranje: vsebuje test od zgornje do spodnje meje,
- testiranje UDP: testira se protokol UDP,
- testiranje TCP: testira se protokol TCP.



Slika 11. Diagram primerov uporabe.

Slika 12 prikazuje diagram pretoka podatkov. Upravitelj vnese nastavitve (parametre) v datoteko »settings« in »testFile«. Pred zagonom testiranja se prenesejo potrebni programi in datoteke na merjence. Sledi zagon testov in po končanem testiranju prenos rezultatov testov na nadzornika.



Slika 12. Diagram podatkovnih tokov.

Program je sestavljen iz dveh delov:

- del, ki je namenjen merjencem: odjemalec, strežnik in usmerjevalnik,
- del, ki je namenjen nadzorniku.

### 3.2.6.1.1. Del namenjen merjencem

Potrebna orodja (datoteke, mape in programi), se nahajajo v mapi »KLIENT«. Vsebina mape:

- qualityTestTCP.sh: skrbi za sprotni vpogled in izpis kvalitete enakih TCP testov,
- qualityTestUDP.sh: skrbi za sprotni vpogled in izpis kvalitete enakih UDP testov,
- runProgramClient.sh: skrbi za vodenje virtualnih oken (ang. screen), na njih zažene program za testiranje,
- runScreen.sh: program, ki zažene virtualno okno,
- runTestClnet.sh: glavni program, na katerem je večina testnih algoritmov,
- scpScriptClient.sh: program, ki ga uporablja samo odjemalec, ta prenese testne rezultate k nadzorniku,
- mapa z nazivom REZULTATI: sem glavni program (runTestClient.sh) shranjuje vse datoteke trenutnega testa.

### 3.2.6.1.2. Del namenjen nadzorniku

Vsebina orodij (datoteke, mape in programi):

- createRules.sh: služi za gradnjo datoteke s pravili, ki jo potrebuje usmerjevalnik,
- runProgram.sh: glavni nadzornikov program, ki nadzira in manipulira teste,
- runTest.sh: izvršni program, s katerim manipulira runProgram.sh,
- scpScript.sh: služi za prenos datotek in programov k merjencem,
- sshScript.sh: zaganja, ustavlja in nastavlja programe na merjencih,
- settings: datoteka, kjer so shranjeni glavni parametri,
- mapa z nazivom KLIENT: tu so programi, datoteke in mape merjencev,
- mapa z nazivom REZULTATI: v to mapo pošlje rezultate testa (služi za testiranje, če je odjemalec že končal s testiranjem),
- mapa z nazivom TESTI: v to mapo se prenesejo vsi testi,
- datoteka testFile – nahaja se v mapi TESTI: v njej so shranjeni načini testiranja,
- createGraph.sh – nahaja se v mapi TESTI: program namenjen risanju grafov.

### 3.2.6.2. Nastavitve programa

Za nemoteno delovanje program potrebuje:

- naslove IP vseh merjencev in nadzornika,
- naslova obeh omrežij,
- nazive mrežnih vmesnikov (npr. eth0) za vse merjence in nadzornika,
- številko vrat merjencev in nadzornika, kjer posluša strežnik SSH,
- geslo za dostop do merjencev in nadzornika,
- velikost največjega in najmanjšega okvirja Ethernet (MTU),
- določiti korak (vrednost, ki se odšteva od največje vrednosti MTU),
- nastavitve načina testiranja.

Vsebina datoteke testFile (Izpis 1. ):

```
#Tu se nastavijo vsi parametri za določeno skupino testov
#st.merjencev (dva,tri) | nacin testa (all,optimal) | fiksno "runProgramClient.sh" |
    fiksno "runTestClient.sh" | testni protokol (UDP,TCP) | interval izpisa v sek.
    npr. 1 | trajanje testa v sek. npr. 3 | ponovitev testa npr. 2
dva 0 all runProgramClient.sh runTestClient.sh UDP 1 3 2
dva 0 all runProgramClient.sh runTestClient.sh TCP 1 3 2
```

Izpis 1. Vsebina datoteke »testfile«.

Vsebina datoteke settings (Izpis 2. ):

```
#V tej datoteki se nastavijo vse glavne nastavitve
#Podatki PRVEGA odjemalca "odjemalec"
ipPrviClient=192.168.0.72
nazivEthernetPrvi=eth0
gesloPrvi=burek
```

portPrvi=22

#Podatki DRUGEGA odjemalca "usmerjevalnik"

ipDrugiClient=192.168.0.71

#Je naziv vmesnika, ki je povezan v omrežje z odjemalcem in nadzornikom

nazivEthernetDrugi1=eth1

#Je naziv vmesnika, ki je povezan v omrežje s strežnikom

nazivEthernetDrugi2=eth0

gesloDrugi=burek

portDrugi=22

#Podatki TRETJEGA odjemalca "strežnik"

ipTretjiClient=

nazivEthernetTretji=eth0

gesloTretji=burek

portTretji=22

#Podatki REMOTE odjemalca "nadzornik"

ipRemote=192.168.0.1

nazivEthernetRemote=eth0

gesloRemote=swdrg21011983

userRemote=dzajc

portRemote=2101

#Spremenljivke za testiranje

#ta naslov je naslov omrežja, kjer so usmerjevalnik, odjemalec in nadzornik

naslovOmrežjaEna=192.168.0.0

#ta naslov je naslov omrežja, kjer sta usmerjevalnik in strežnik

naslovOmrežjaDva=192.168.2.0

#opis ki služi kot orientacijo

opisTesta="Povezava LAN, oba kabelj"

maxMTU=1500

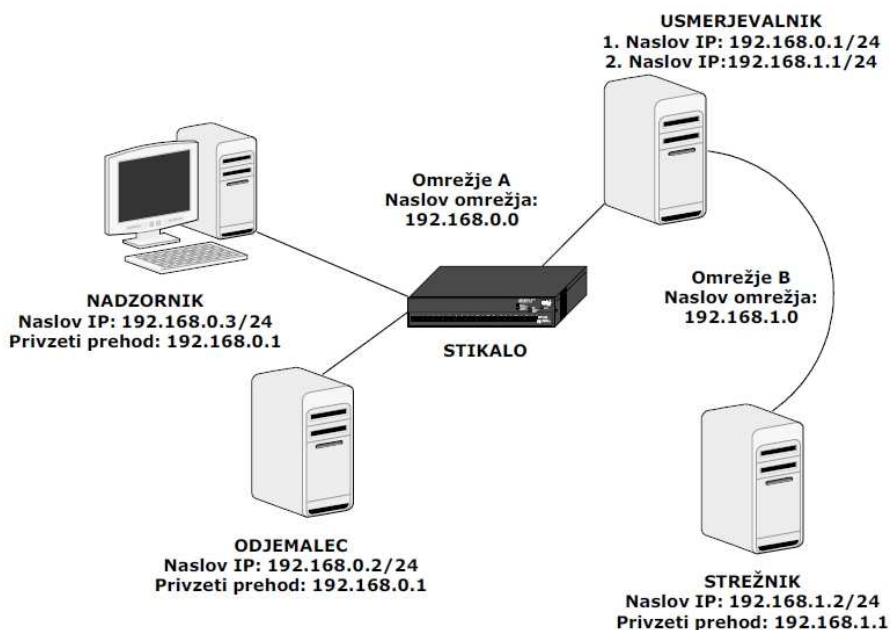
korakMTU=50

minMTU=88

lanSpeed=100

Izpis 2. Vsebina datoteke »settings«.

### 3.2.6.3. Nastavitve merjencev



Slika 13. Povezava merjencev in nadzornika, ter njihovi naslovi IP, maska omrežja in privzeti prehod.

Merjenci morajo biti povezani tako, kakor prikazuje slika 13. Odjemalec, usmerjevalnik in nadzornik so povezani v isto omrežje, strežnik pa je povezan na usmerjevalnik. Naslovi IP so lahko nastavljeni kot na sliki 13:

- usmerjevalnik ima na enem vmesniku naslov IP 192.168.0.1/24 (/24 – notacija CIDR), na drugem pa 192.168.1.1/24 (tisti, ki je povezan s strežnikom),
- nadzornik 192.168.0.3/24,
- odjemalec 192.168.0.2/24,
- strežnik 192.168.1.2/24.

Privzeti prehodi (Slika 13):

- odjemalec 192.168.0.1,
- nadzornik 192.168.0.1,
- strežnik 192.168.1.1.

Poleg tega, kar nam že sama slika prikazuje, moramo še:

- zagnati sshd (/etc/init.d/sshd start),
- ustaviti NetworkManager (/etc/init.d/NetworkManager stop).

### 3.2.6.4. Prenašanje programov in datotek

S pomočjo programa »scpScript.sh« nadzornik pošlje vse potrebne programe in datoteke na merjence. To je expect-ov program, ki se s pomočjo odjemalca scp poveže z merjencem in prenese preko protokola SSH [14] (ang. Secure SHell – varnostne lupine) datoteke. Obratno pot prenosa (od odjemalca k nadzorniku) opravlja drugi program »scpScriptClient.sh«, ki ima podobne lastnosti z nekaterimi dodatki.

### 3.2.6.5. Povezava in zagon skript

Program uporablja v ta namen program »sshScript.sh«. To je expect-ov program, ki se s pomočjo odjemalca SSH poveže z merjenci in zažene program »runProgramClient.sh« s parametri, ki jih potrebujejo. V primeru, da je merjenec usmerjevalnik, neposredno uporabi program »runTestClient.sh« in se tako izogne nepotrebnemu zagonu programa »runScreen.sh«.

### 3.2.6.6. Načini testiranja in delovanje

S testnim programom lahko testiramo hitrost povezave s protokoloma TCP in UDP. To lahko storimo na dva načina:

- Optimal (test se izvaja do meje, ki je še smotrna):
  - TCP: najmanjša velikost sprejemnega vmesnika, ki ne vpliva na hitrost,
  - UDP: največja hitrost, kjer ni izgub.
- All (test, kjer se parametri spreminjajo konstantno ne glede na rezultate):
  - TCP: spreminjanje velikosti sprejemnega okna od največje proti najmanjši vrednosti s konstanto, ki je določena vnaprej (vrednost konstante je 1),
  - UDP: se spreminja hitrost oddajanja (vrednost konstante je 10% teoretične hitrosti povezave).

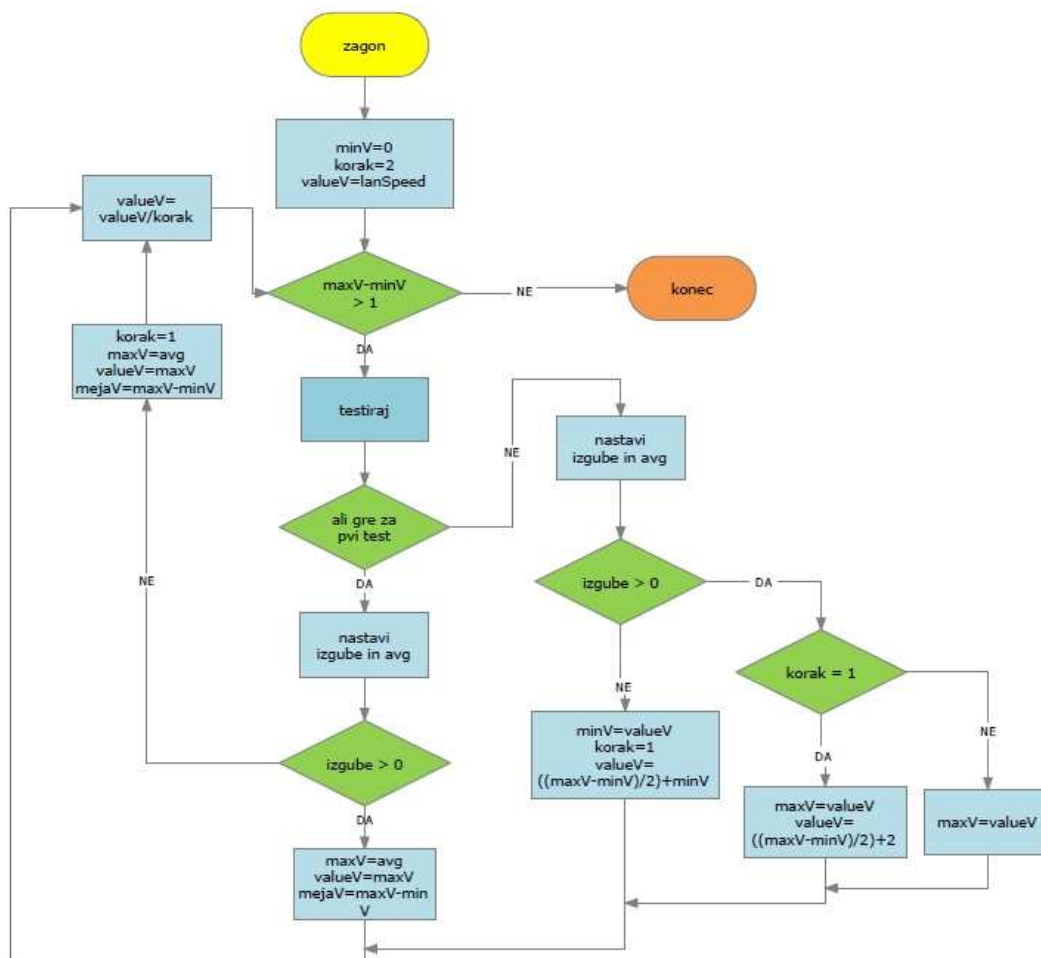
#### 3.2.6.6.1. Optimalno testiranje UDP

Program potrebuje za optimalno testiranje UDP-ja naslednje spremenljivke:

- izgube: izgube v testih,
- avg: povprečna hitrost testov,
- maxV: zgornja meja testne hitrosti,
- minV: spodnja meja testne hitrosti (na začetku je 0),
- korak: vrednost, ki spreminja vrednost spremenljivke valueV,
- valueV: testna hitrost,
- mejaV: razlika med maxV in minV.



Postopek testiranja poteka kot prikazuje diagram poteka (Slika 14).



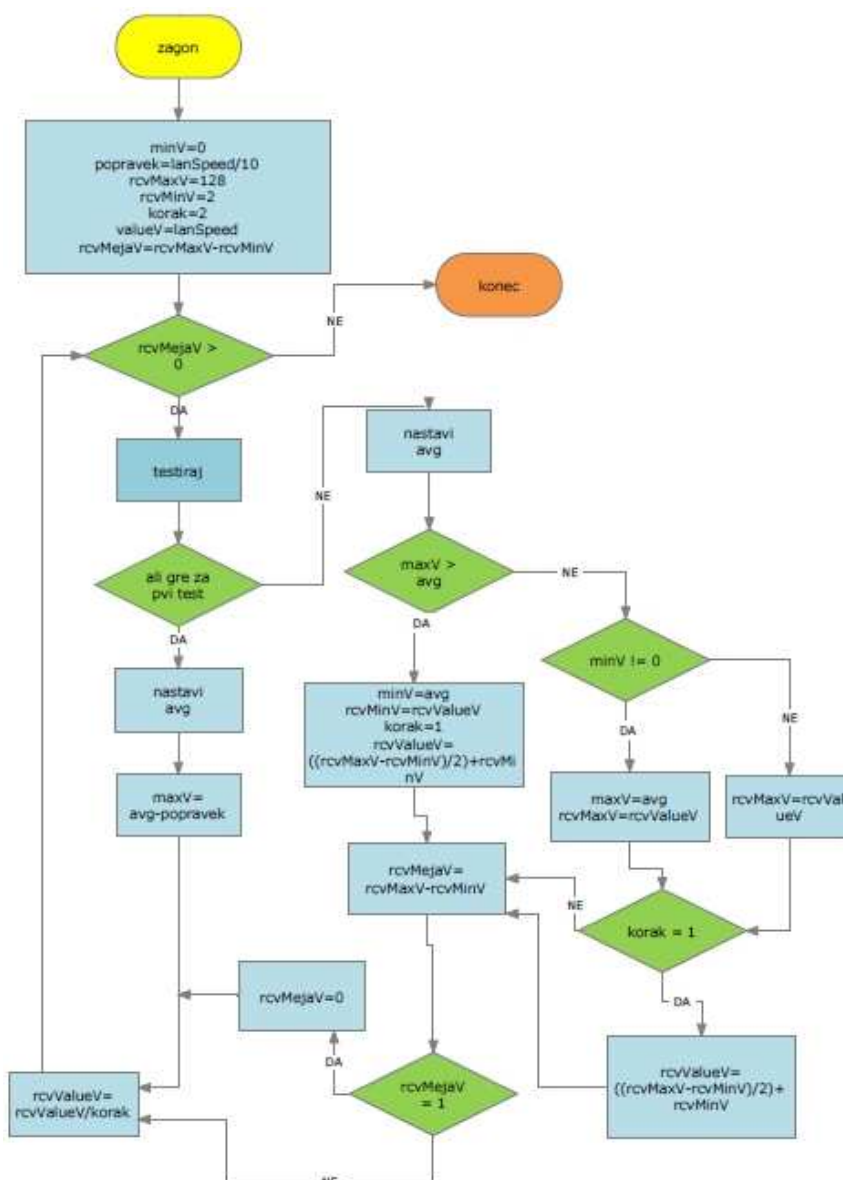
Slika 14. Diagram poteka optimalnega testiranja UDP.

### 3.2.6.6.2. Optimalno testiranje TCP

Program potrebuje za optimalno testiranje TCP-ja naslednje spremenljivke:

- popravek: 10% manjša hitrost kot je teoretična hitrost omrežja,
- avg: povprečna hitrost testov,
- maxV: zgornja meja testne hitrosti,
- minV: spodnja meja testne hitrosti (na začetku je 0),
- rcvMaxV: zgornja meja sprejemnega okna,
- rcvMinV: spodnja meja sprejemnega okna,
- rcvMejaV: zaustavitveni pogoj (razlika med rcvMaxV in rcvMinV),
- korak: vrednost, ki spreminja vrednost spremenljivke valueV,
- rcvValueV: testna vrednost sprejemnega okna.

Postopek testiranja poteka kot prikazuje diagram poteka (Slika 15).



Slika 15. Diagram poteka optimalnega testiranja TCP.

### 3.2.6.7. Shranjevanje testnih datotek

Pri shranjevanju testnih datotek naletimo na tri različne oblike:

- testne datoteke (vsebujejo rezultate enega testa):  
»client,2009-09-21,23-10-22.test«,
- zapakirane testne datoteke (vsebuje testne datoteke):  
»client,2009-09-25,23-10-22.tar«,
- končne datoteke (vsebuje zapakirane testne datoteke):  
»client,2009-09-21,33-10-22.TCP.tar« ali »client,2009-09-21,43-10-22.UDP.tar«.

Vsak opravljen test dobi svoje ime, ki je sestavljeno iz predpone, datuma, časa in končnice (npr. »client,2009-09-10,23-21-03.test«). Enakih testov je lahko več (testi z istimi parametri »test«), vendar vsak dobi enolično ime. Ko se končajo enolični testi, se shranijo podatki o testih v datoteko »rezSlika.slika«. Nato se vse datoteke zapakirajo v »tar«, razen »rezSlika.slika«. Po zaključku skupine testov se datoteka »rezSlika.slika« in »tar« datoteke zapakirajo v eno samo datoteko. Ta ima končnico »UDP.tar« ali »TCP.tar«, ki se jo pošlje k nadzorniku.

### 3.2.6.8. Risanje grafov

Program uporablja za risanje grafov ločen program. Le-ta iz končne datoteke rezSlika.slika pridobi podatke, ki jih sortira po vrsti in jih nato nariše s programom Gnuplot, ki je opisan v točki 3.2.5. Program je napisan v datoteki z imenom »createGraph.sh«.

## 4. REZULTATI TESTIRANJA IN UGOTOVITVE

Za testiranje 100 Mbit/s omrežja sem uporabil naslednje računalnike za merjenje:

- odjemalec:
  - procesor: Celeron 533 MHz,
  - delovni pomnilnik (RAM): 224 MB,
  - mrežni vmesnik: 10/100 Mbit/s (Edimax IEEE-802.3 10/100Base-TX).
- strežnik:
  - procesor: Celeron 850 MHz,
  - delovni pomnilnik (RAM): 384 MB,
  - mrežni vmesnik: 10/100 Mbit/s (Edimax IEEE-802.3 10/100Base-TX).
- usmerjevalnik:
  - procesor: Athlon 2000 MHz,
  - delovni pomnilnik (RAM): 512 MB,
  - mrežni vmesnik: 2x 10/100 Mbit/s (Edimax IEEE-802.3 10/100Base-TX).

Stikalo:

- model: Jaht js-2016EP Dual-Speed Switch Hub 16-port,
- hitrost: 10/100 Mbit/s.

Za testiranje 1 Gbit/s omrežja sem uporabil naslednje računalnike za merjenje:

- odjemalec:
  - procesor: Intel Xeon X3220 (Quad Core), 2,4 GHz,
  - delovni pomnilnik (RAM): 8 GB,
  - mrežni vmesnik: Broadcom Corporation NetXtreme BCM5721 Gigabit.
- strežnik:
  - procesor: Intel Xeon X3220 (Quad Core), 2,4GHz,
  - delovni pomnilnik (RAM): 8 GB,
  - mrežni vmesnik: Broadcom Corporation NetXtreme BCM5721 Gigabit.

- usmerjevalnik:
  - procesor: Intel Xeon X3220 (Quad Core), 2,4 GHz,
  - delovni pomnilnik (RAM): 8 GB,
  - mrežni vmesnik: 2x Broadcom Corporation NetXtreme BCM5721 Gigabit.

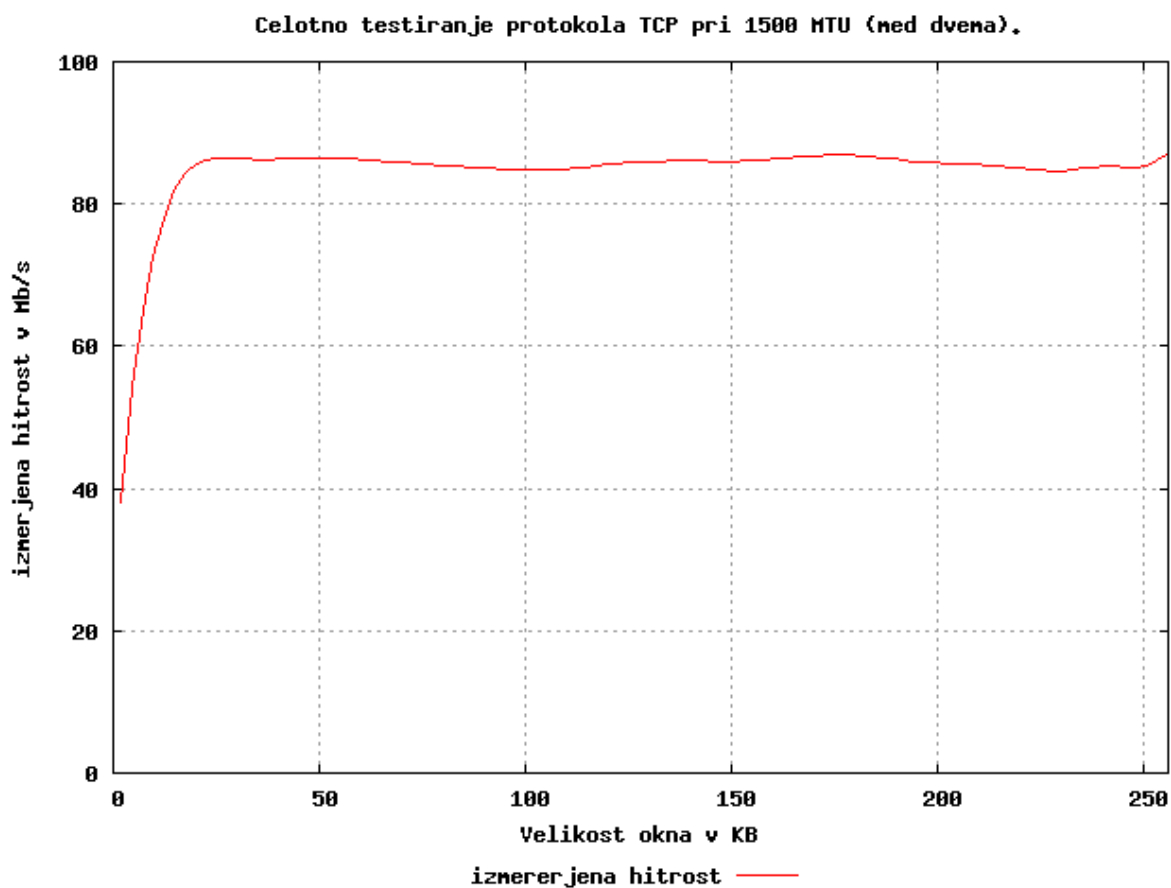
Stikalo:

- model: Dell PowerConnect 5448,
- hitrost: 10/100/1000 Mbit/s.

#### 4.1. Rezultati

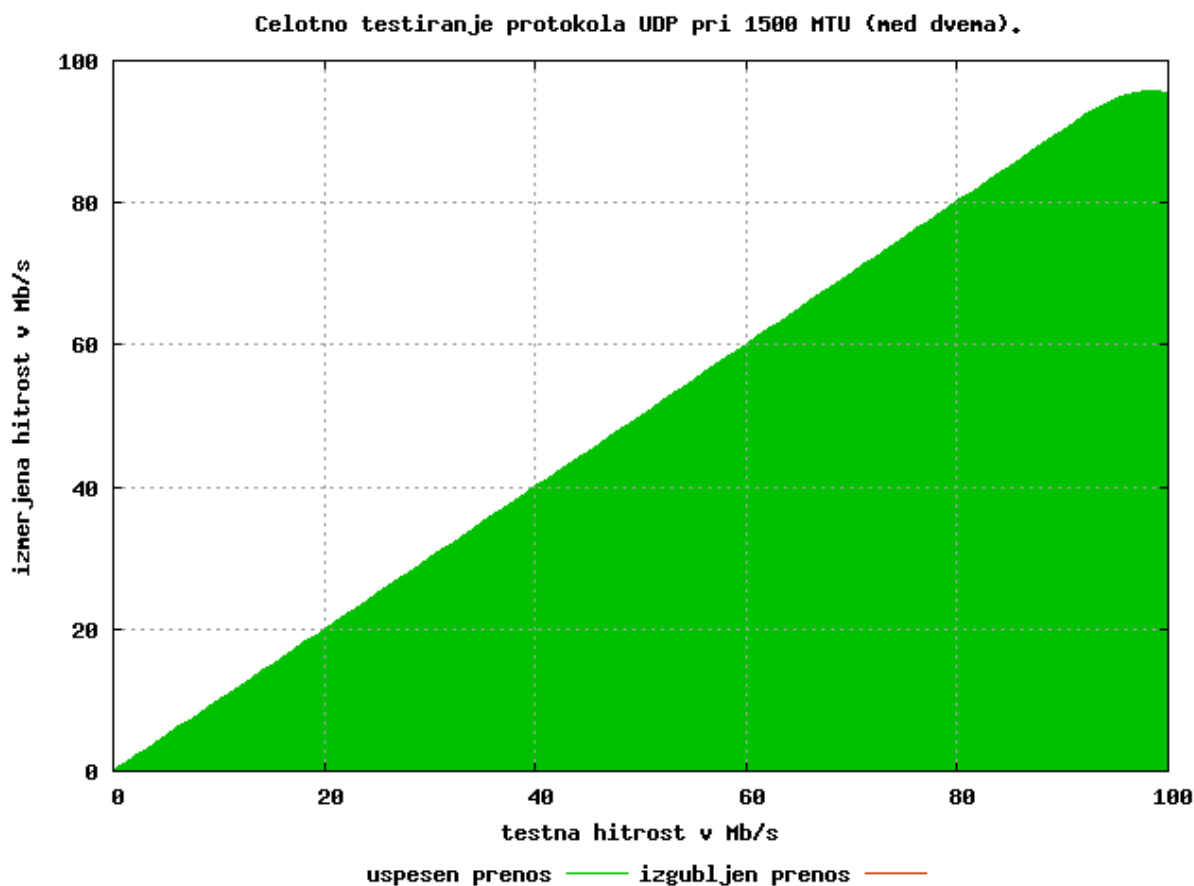
##### 4.1.1. Omrežje 100 Mbit/s

Iz grafa (Slika 16) je razvidno, da povečanje velikosti okna nad 30 KB ne pripomore k hitrejšemu prenosu.



Slika 16. Celotno testiranje 100 Mbit/s omrežja, med dvema merjencema, pri odvisnosti velikosti okna s protokolom TCP, 1500 B MTU in brez pravil v požarnem zidu.

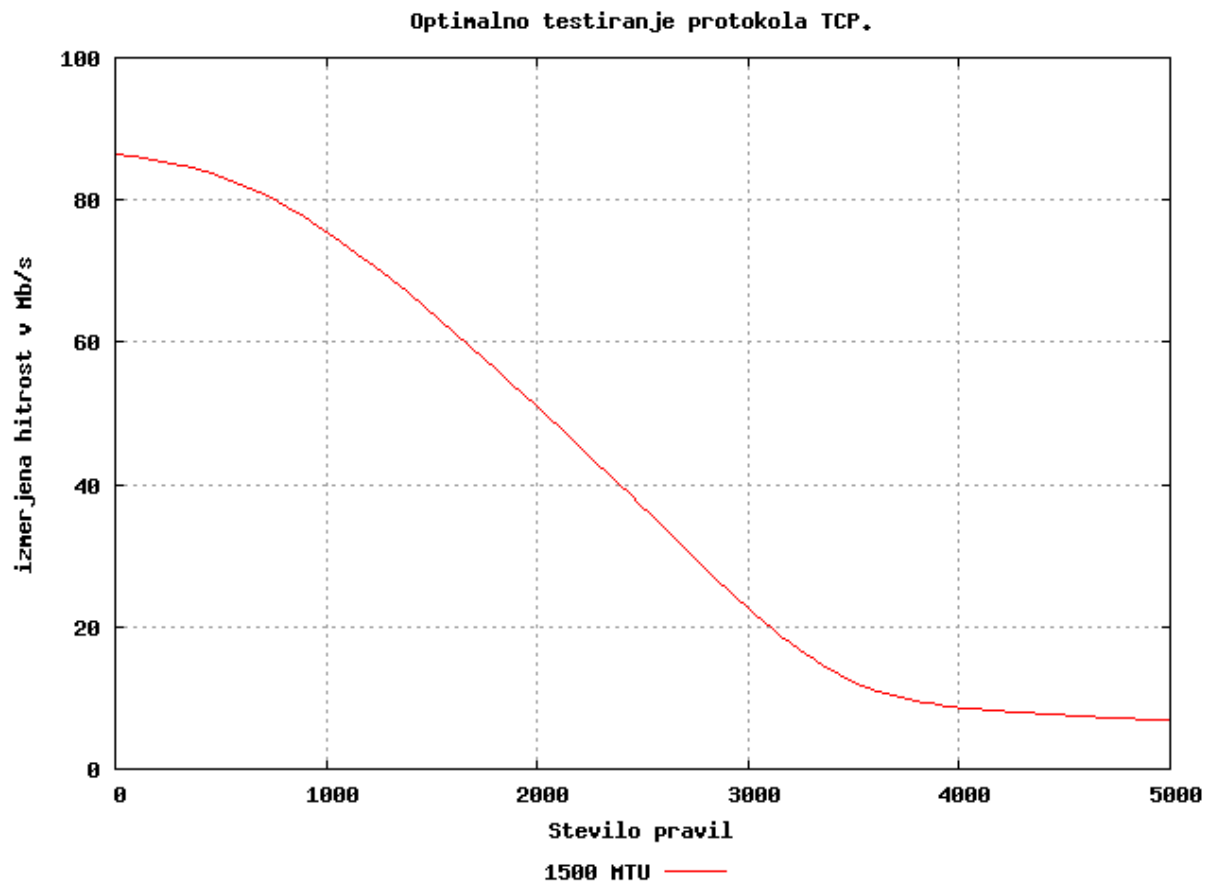
Iz grafa (Slika 17) je razvidno, da povečanje hitrosti oddajanja ne vpliva na izgubo prenosa (glej za primerjavo Slika 24). Vidimo tudi, da odjemalec ne zmore ustvarjati pakete hitreje od 95 Mbit/s.



Slika 17. Celotno testiranje 100 Mbit/s omrežja, med dvema merjencema, pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in z nobenim pravilom v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete).

Iz grafa (Slika 18) je razvidno:

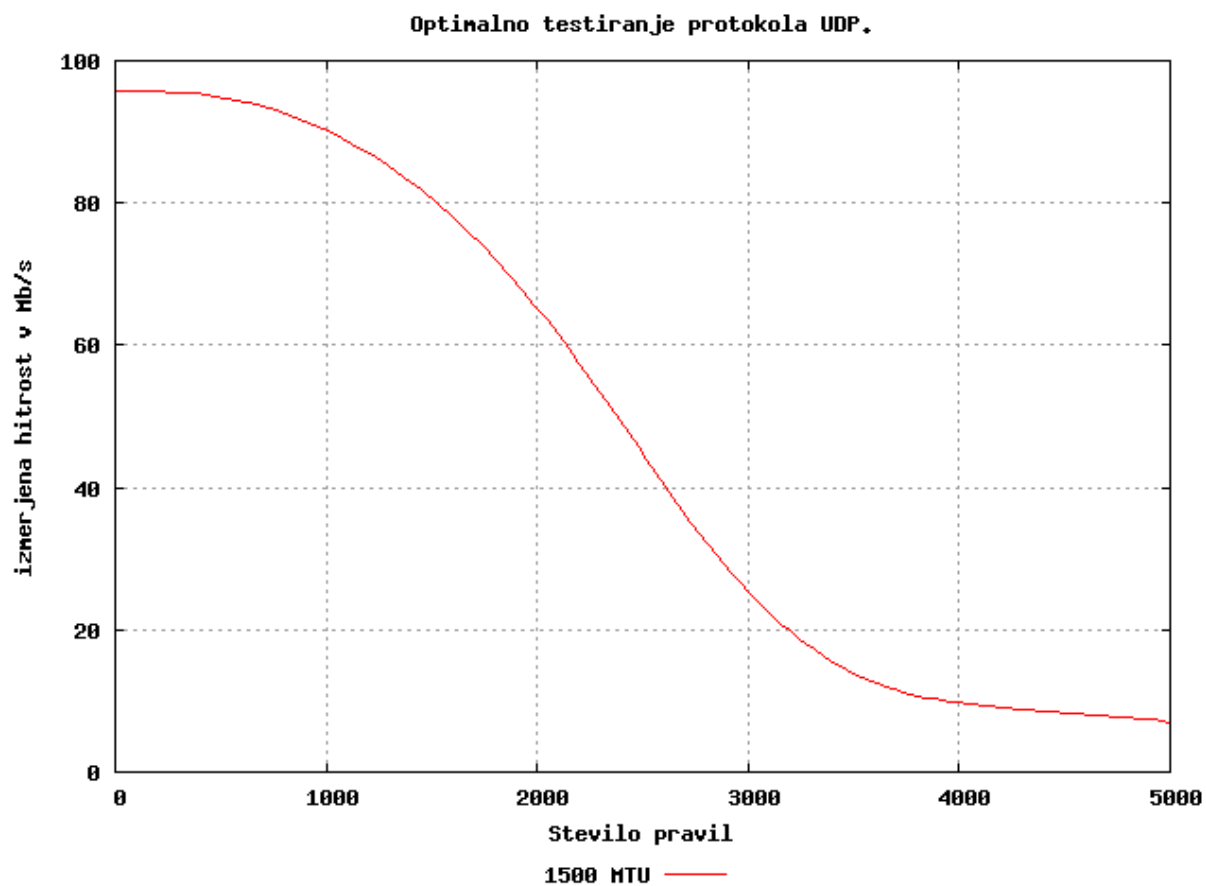
- območje pravil med 0 in 500: ni vpliva na hitrost prenosa,
- območje pravil med 500 in 4000: eksponentni padec hitrosti prenosa,
- območje pravil med 4000 in 5000: enakomerna hitrost.



Slika 18. Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri 1500 B MTU (korak 500).

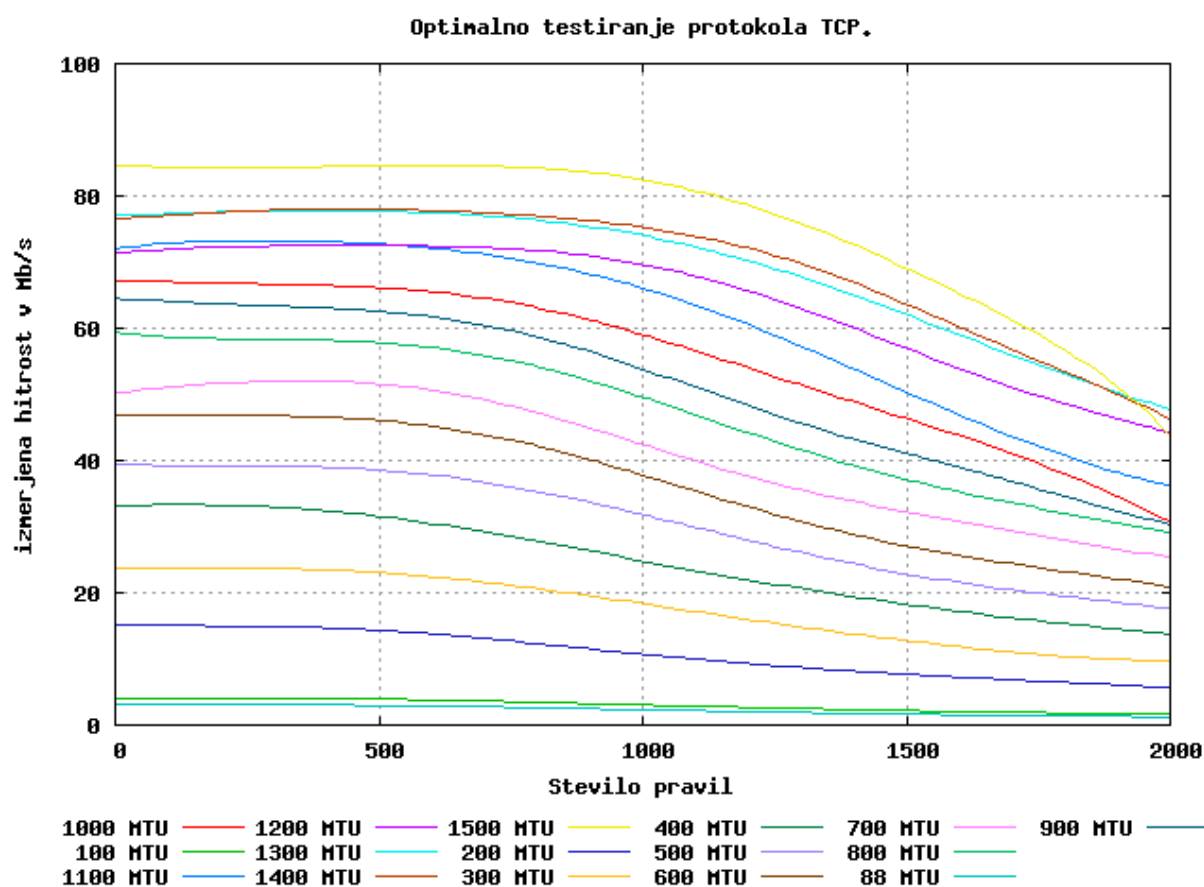
Iz grafa (Slika 19) je razvidno:

- območje pravil med 0 in 1000: ni vpliva na hitrost prenosa,
- območje pravil med 1000 in 4000: eksponentni padec hitrosti prenosa,
- območje pravil med 4000 in 5000: enakomerna hitrost.



Slika 19. Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri 1500 B MTU (korak 500).

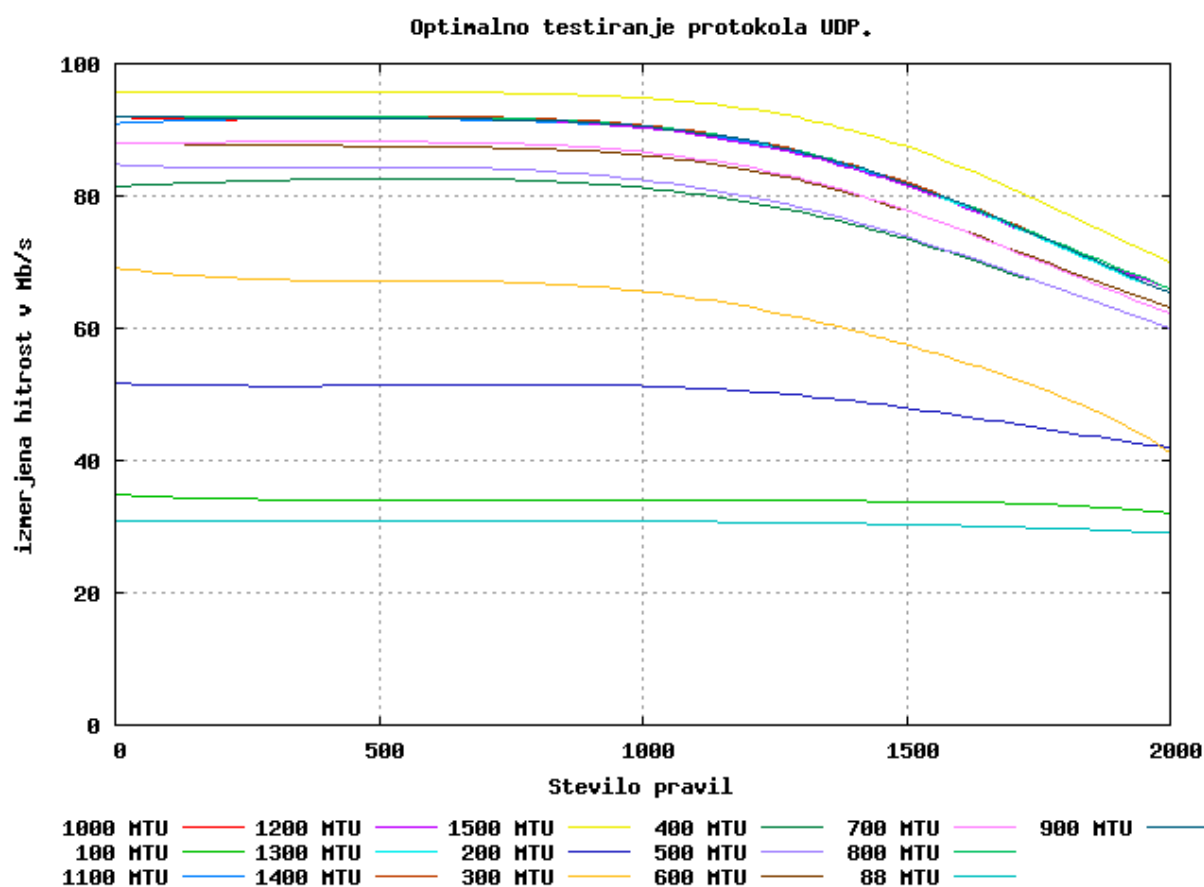
Iz grafa (Slika 20) je razvidno, da se z manjšanjem velikosti MTU spreminja hitrost prenosa, posledično pa tudi meja eksponentnega padca. Vidimo tudi, da pri velikosti MTU 88 in 100, hitrost ostaja za vse korake skorajda enaka.



Slika 20. Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri različnih velikostih MTU.

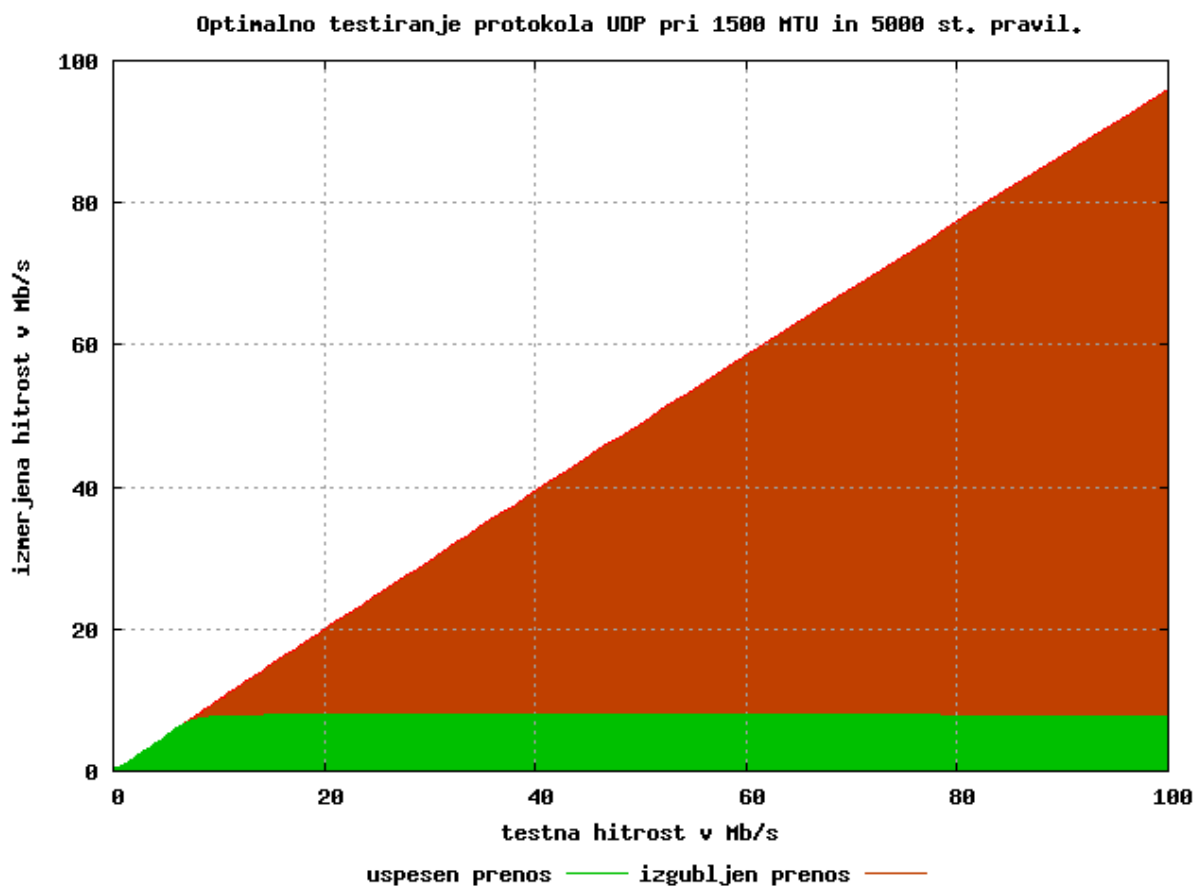


Iz grafa (Slika 21) je razvidno, da se z manjšanjem velikosti MTU spreminja hitrost prenosa, posledično pa tudi meja eksponentnega padca. Vidimo tudi, da pri velikosti MTU 88 in 100, hitrost ostaja za vse korake skorajda enaka.



Slika 21. Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri različnih velikostih MTU.

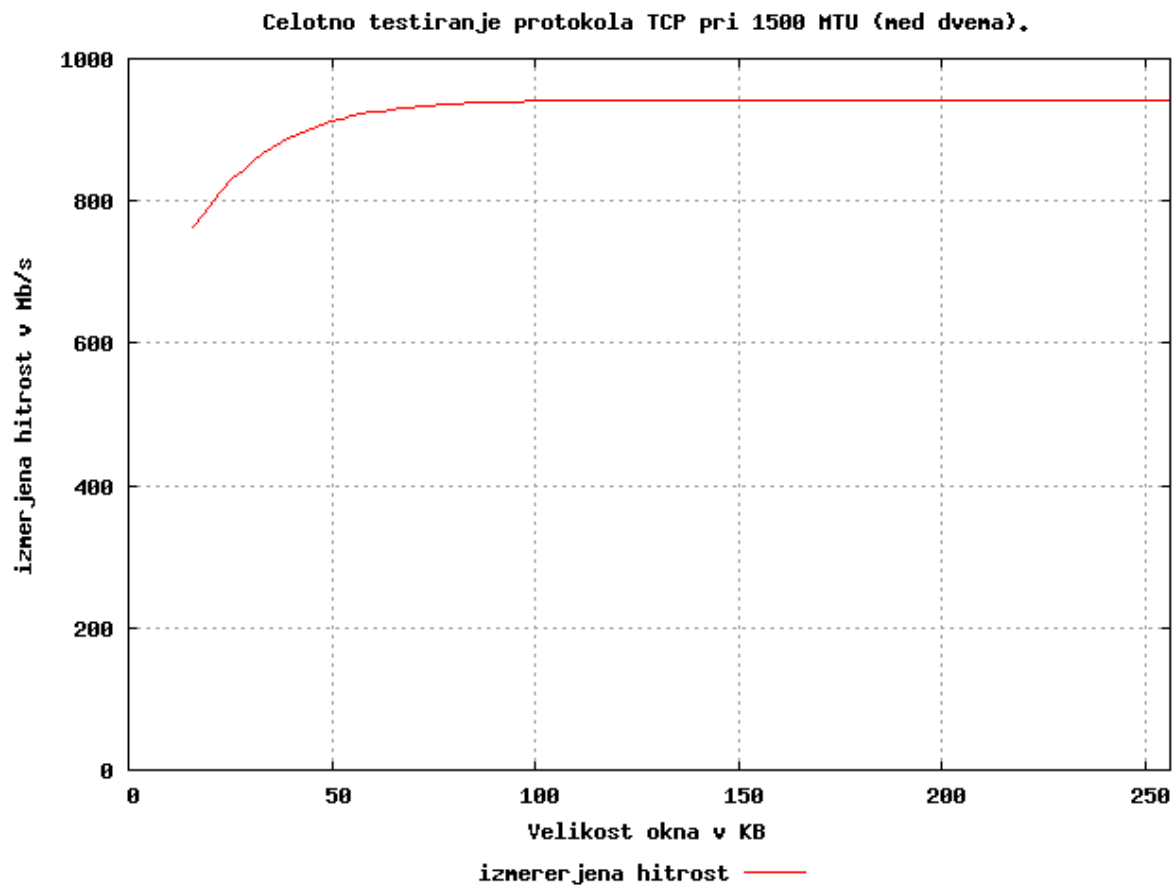
Iz grafa (Slika 22) je razvidno, da se že pri povečanju hitrosti oddajanja nad 8 Mbit/s, zaradi velikega števila pravil v požarnem zidu (5000), prične prenos izgubljati. Razlogi za to so opisani v točki 2.3.1.3.



Slika 22. Celotno testiranje 100 Mbit/s omrežja pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in 5000 pravili v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete).

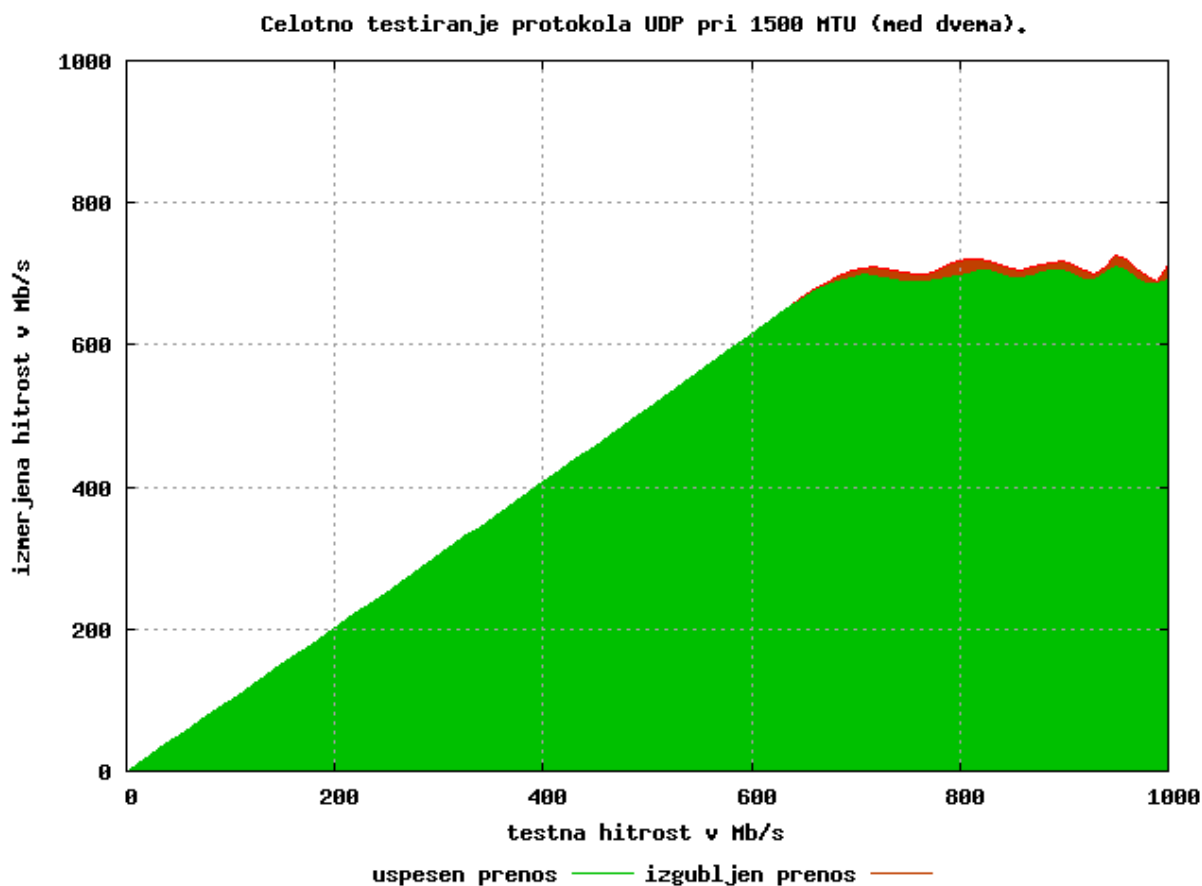
#### 4.1.2. Omrežje 1 Gbit/s

Iz grafa (Slika 23) je razvidno, da povečanje velikosti okna nad 80 KB ne pripomore k hitrejšemu prenosu.



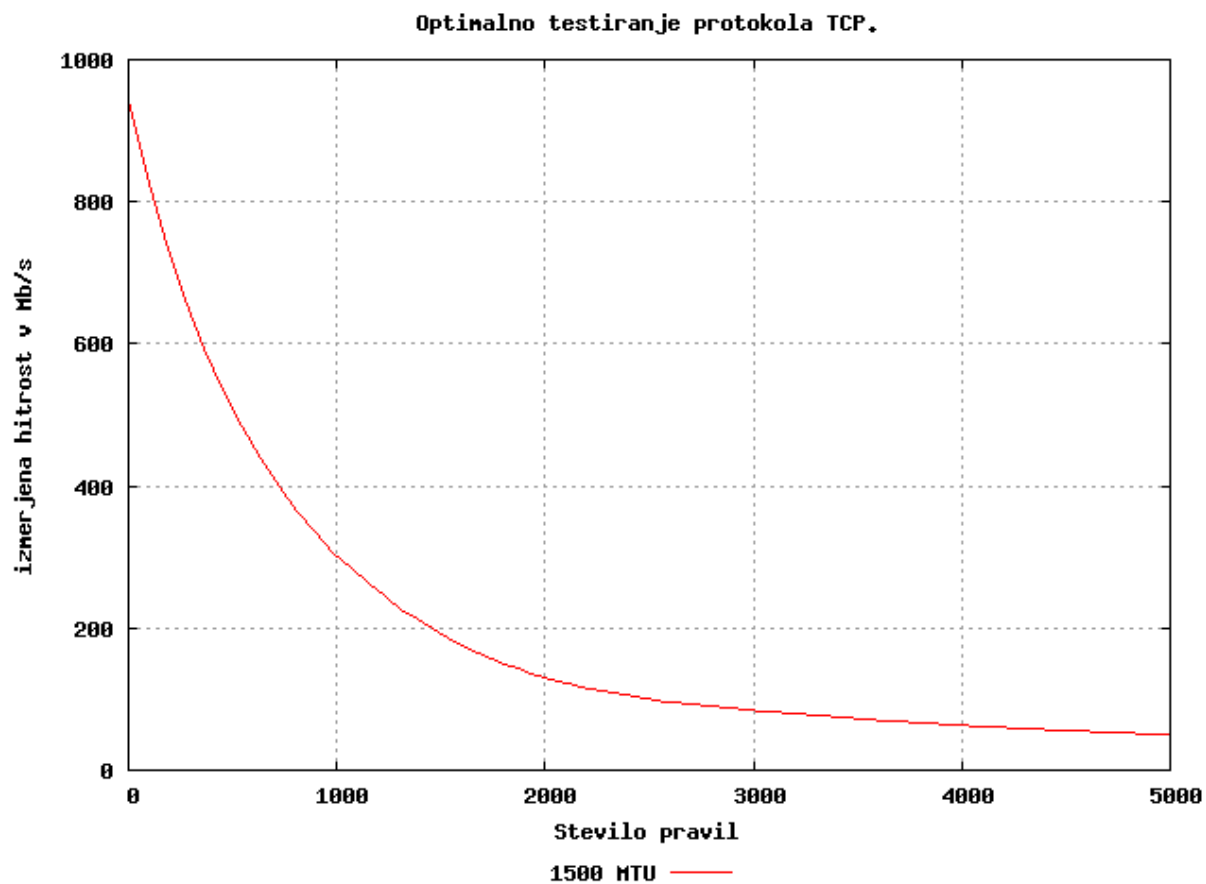
Slika 23. Celotno testiranje 1 Gbit/s omrežja, med dvema merjencema, pri odvisnosti velikosti okna s protokolom TCP, 1500 B MTU in brez pravil v požarnem zidu.

Iz grafa (Slika 24) je razvidno, da se pri povečanju hitrosti oddajanja nad 650 Mbit/s prične izgubljati prenos. Razlogi za to so opisani v točki 2.3.1.3. Na grafu vidimo, da odjemalec ne zmore ustvarjati pakete hitreje od 750 Mbit/s.



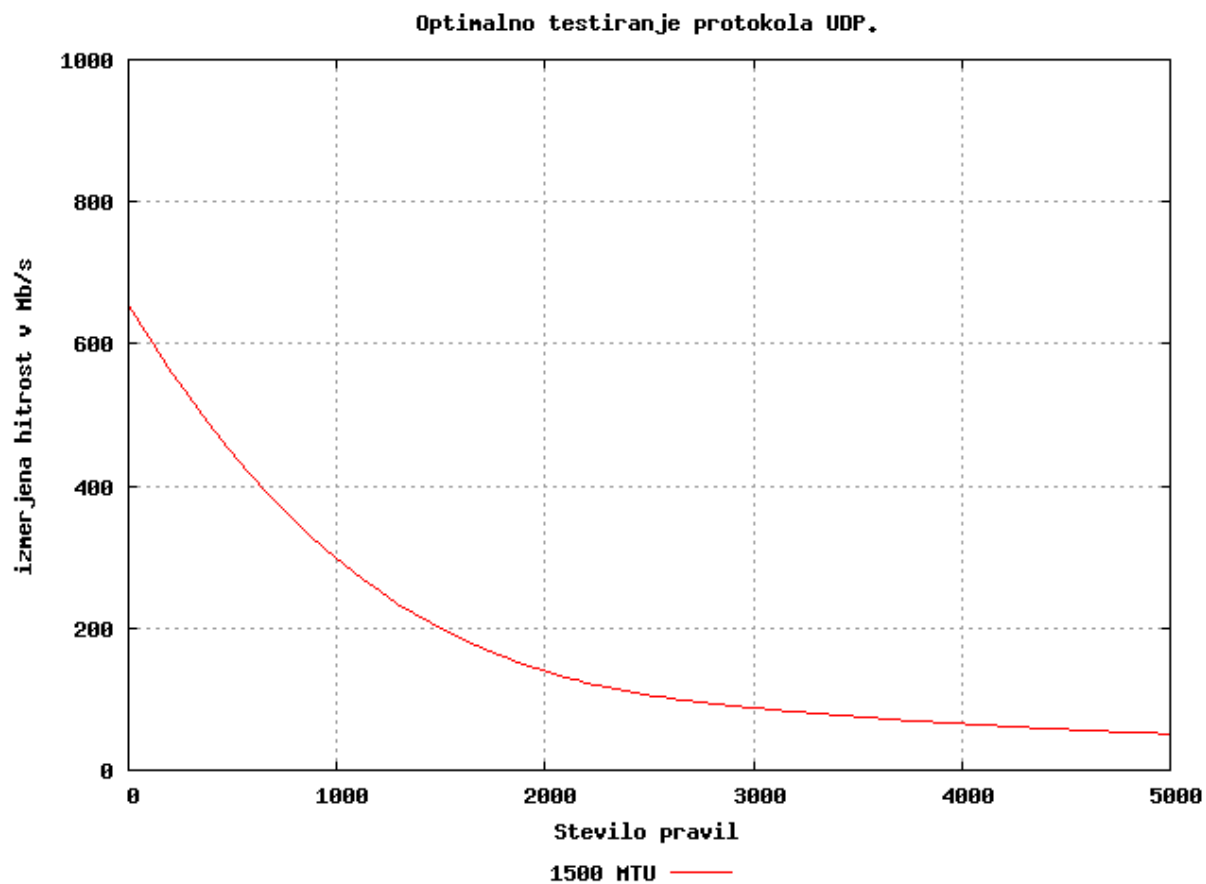
Slika 24. Celotno testiranje 1 Gbit/s omrežja, med dvema merjencema, pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in z nobenim pravilom v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete).

Iz grafa (Slika 25) je razvidno, da z večanjem števila pravil v požarnem zidu do meje 3000 posledično eksponentno zmanjšujemo hitrost prenosa.



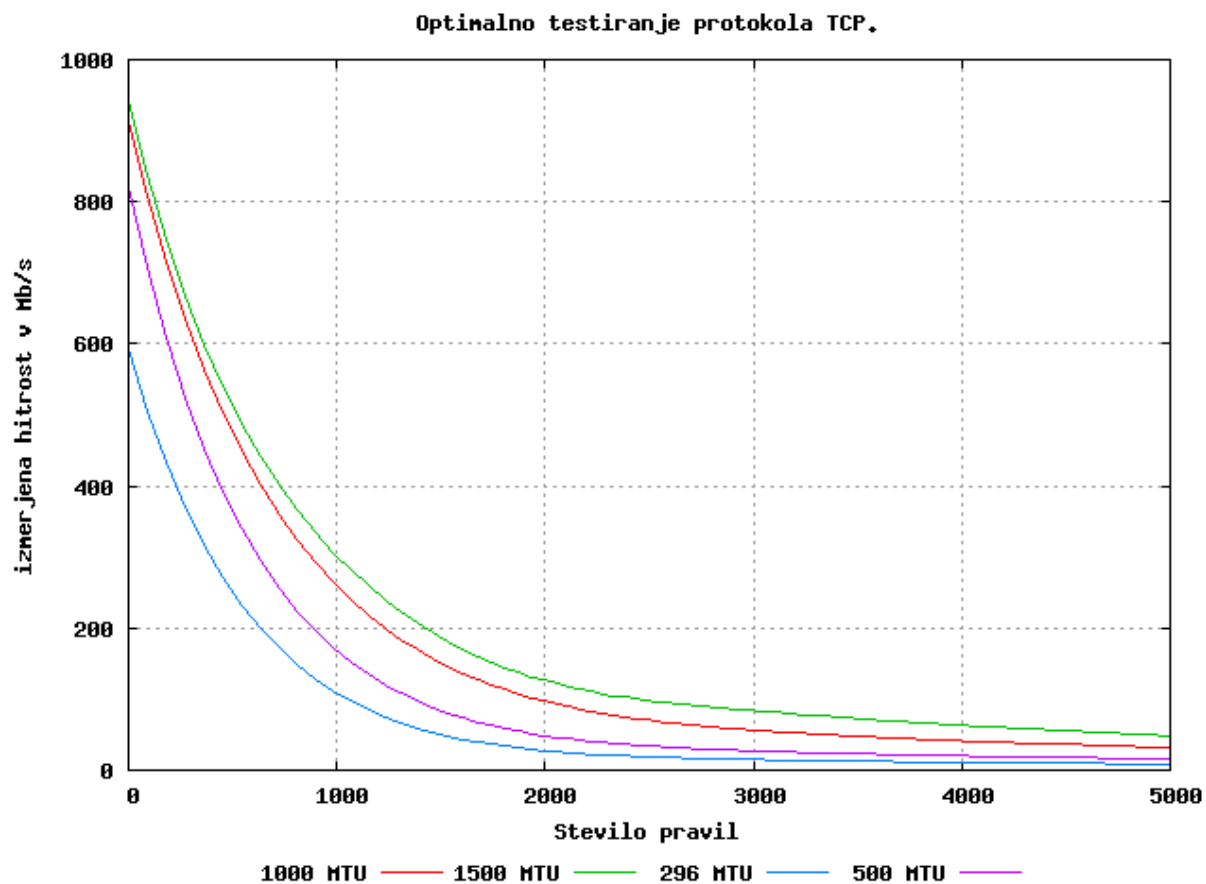
Slika 25. Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri 1500 B MTU (korak 500).

Iz grafa (Slika 26) je razvidno, da se z večanjem števila pravil v požarnem zidu do meje 3000 eksponentno zmanjšuje hitrost prenosa.



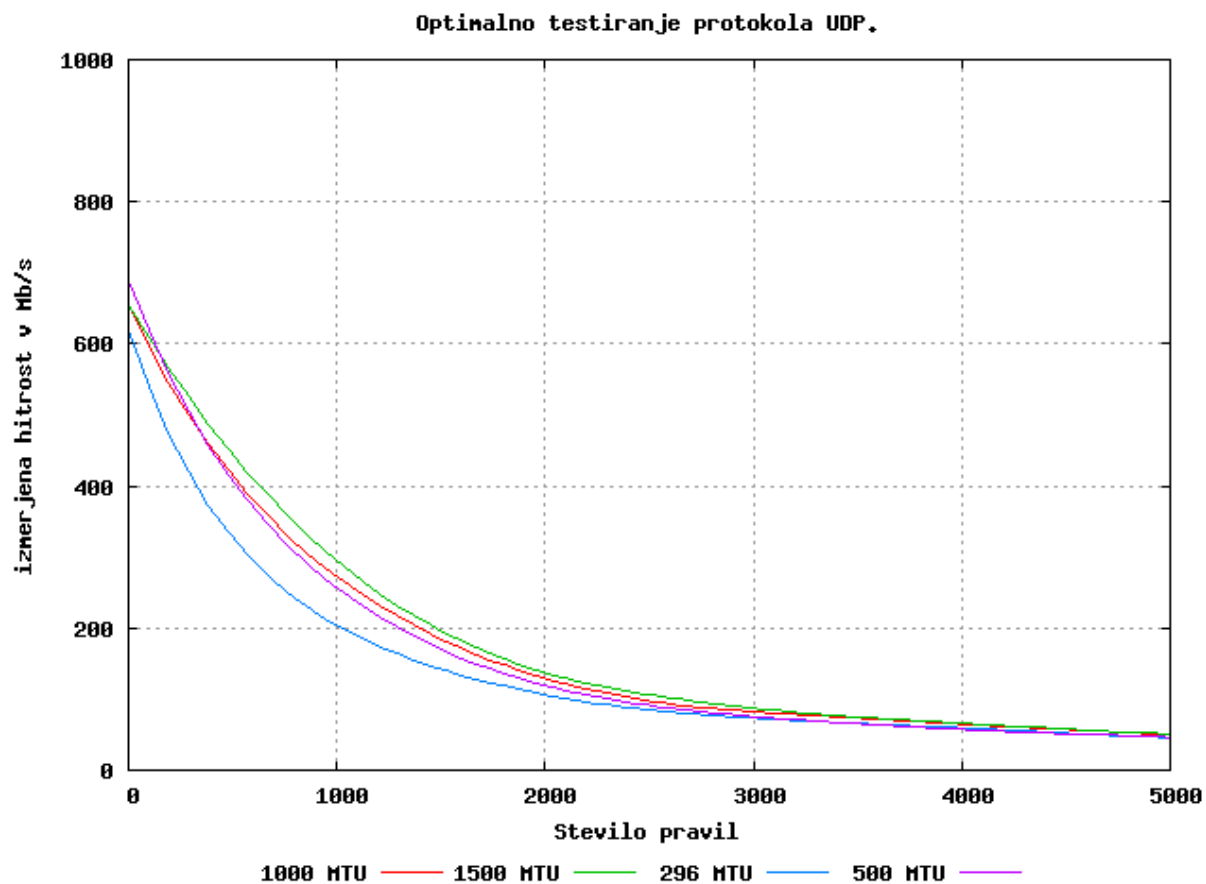
Slika 26. Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri 1500 B MTU (korak 500).

Iz grafa (Slika 27) je razvidno, da se z manjšanjem velikosti MTU spreminja hitrost prenosa, posledično pa tudi meja eksponentnega padca.



Slika 27. Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri različnih velikostih MTU.

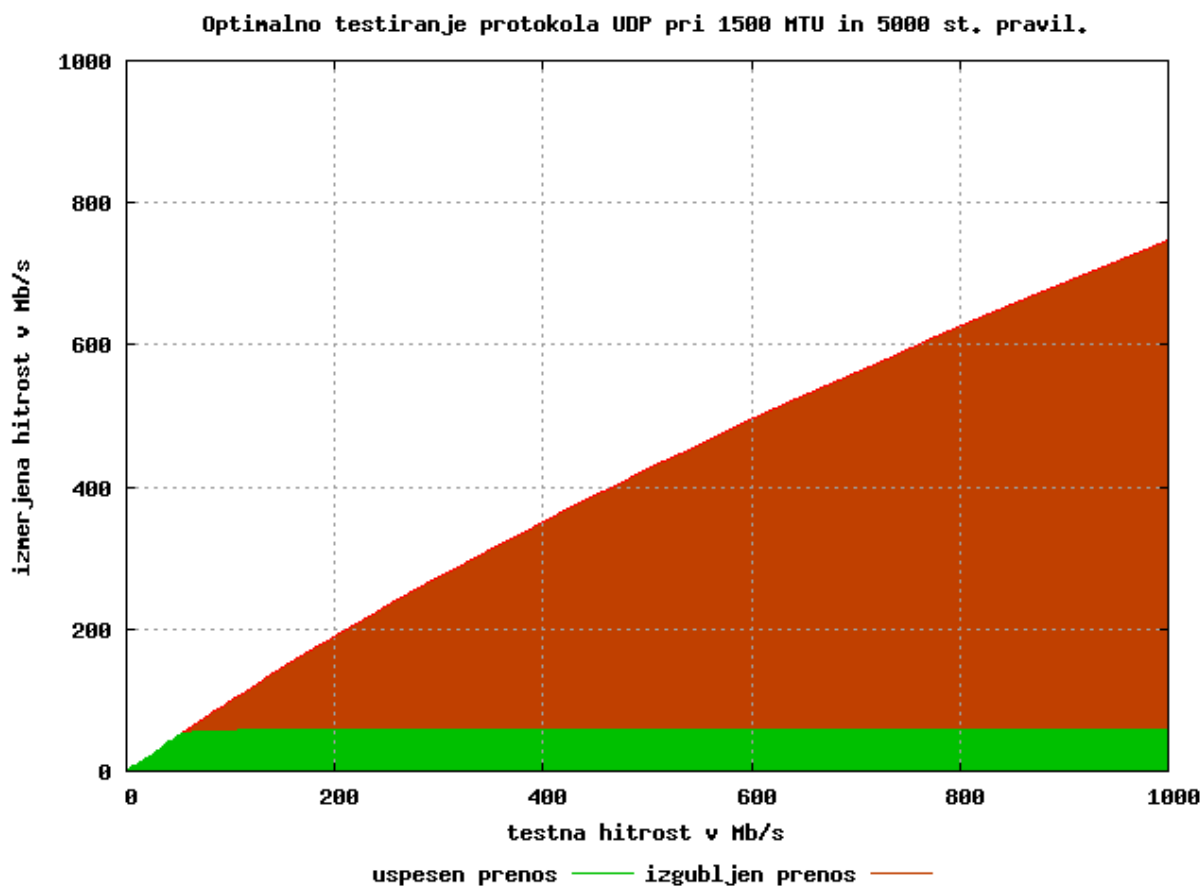
Iz grafa (Slika 28) je razvidno, da se z manjšanjem velikosti MTU spreminja hitrost prenosa, posledično pa tudi meja eksponentnega padca.



Slika 28. Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri različnih velikostih MTU.



Iz grafa (Slika 29) je razvidno, da se že pri povečanju hitrosti oddajanja nad 50 Mbit/s, zaradi velikega števila pravil v požarnem zidu (5000), prične prenos izgubljati. Razlogi za to so opisani v točki 2.3.1.3.



Slika 29. Celotno testiranje 1 Gbit/s omrežja pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in 5000 pravili v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete).

### 4.1.3. Ugotovitve

V času izdelave diplomskega dela sem ugotovil:

- Na 1 Gbit/s omrežju računalniki oz. mrežni vmesniki ustvarjajo pakete UDP počasneje kot TCP (pri 100 Mbit/s omrežju je ravno obratno). Razlog hitrejšega ustvarjanja paketov TCP je, da pri tem pomaga sam mrežni vmesnik (ang. TCP offloading engine – TOE), [23]
- z večanjem velikosti okvirja Ethernet se povečuje tudi hitrost prenosa in posledično tudi meja za optimalen vnos pravil v požarni zid,
- program SSH za svoje delovanje potrebuje velikost okvirja Ethernet 88 B ali več. Razlog za to je, da se segmenti TCP pri vzpostavitvi seje SSH ne smejo fragmentirati (zastavica DF v glavi IP nastavljena na 1),
- pri testiranju na 1 Gbit/s omrežju sem uporabil velikosti okvirja Ethernet 296 B in več: nadaljnje zmanjševanje MTU ni bilo uspešno, saj vzpostavljanje seje SSH z drugimi merjenci ni uspelo iz enakega razloga kot v prejšnji točki, le da smo na teh računalnikih uporabljali daljše ključe RSA, v primeru, da je seja že vzpostavljena, ob spremembi velikosti okvirja, komunikacija normalno poteka,
- ugotovitev na omrežju 1 Gbit/s: v primeru uporabe pravil v požarnem zidu, se moramo zavedati, da nam 500 pravil v požarnem zidu zmanjša hitrost prenosa podatkov za približno 50 odstotkov,
- pri testiranju protokola UDP sem ugotovil, da ob zasičenju omrežja prenos ostane enak ne glede na to, kako hitro se ustvarjajo paketi.

## 5. ZAKLJUČEK

Za diplomsko delo sem moral narediti program, ki samodejno testira hitrost prenosa podatkov po omrežju. Pri izdelavi sem se soočil z nekaterimi težavami:

- kako prenesti skripte in zagnati program na merjencih, kar sem rešil tako, da sem si pomagal s programom Expect,
- ugotovil sem, da se ob odjavi iz merjencev ugasnejo tudi programi, ki sem jih zagnal. Za odpravo te težave sem uporabil program Screen,
- po prototipni izdelavi testnega programa so sledila prva testiranja. Po nekaj urnem čakanju sem ugotovil, da trenutni testi trajajo predolgo. Program sem spremenil tako, da išče mejo na optimalen način,
- zahteva je bila, da mora program omogočati izris grafov. Da sem se izognil kasnejšemu iskanju spremenljivk po testnih datotekah, sem program napisal tako, da je že med testiranjem zabeležil spremenljivke v ločeno datoteko. Na ta način sem si olajšal pisanje programa za izdelavo grafa.

Pri testiranju usmerjevalnikov sem pravilo »ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED« dal na konec in s tem dosegel, da so vsi paketi na usmerjevalniku vedno znova prečili vsa pravila v požarnem zidu. To pravilo je v praksi ponavadi prvo, ker pa sem hotel ugotoviti, kaj se zgodi z usmerjanjem velikega števila kratkih sej, sem to pravilo dal na konec.

Za testiranje hitrosti usmerjanja sem uporabil dva različna tipa računalnikov - zmogljive strežnike in starejše namizne računalnike. Razlika med njimi je bila v procesorju, delovnem pomnilniku in hitrosti omrežnega vmesnika. V obeh okoljih sem izvedel teste, najprej z nobenim pravilom v požarnem zidu in nato s 500 pravili. Iz testnih rezultatov je razvidno:

- v 100 Mbit/s omrežju se hitrost prenosa, s protokolom TCP in UDP, skorajda ne spremeni,
- v 1 Gbit/s omrežju se hitrost prenosa s protokolom TCP zmanjša za približno polovico in s protokolom UDP za približno tretjino.

Na podlagi rezultatov meritev lahko sklepam:

- star računalnik bo zadoščal za manjše 100 Mbit/s omrežje, priklopljeno v internet preko npr. ADSL,
- za večja omrežja, ki potrebujejo kompleksnejši (zapleten oz. z velikim številom pravil) požarni zid, pa tudi zmogljiv strežnik ne zmore dovolj hitro filtrirati prometa. Za takšno omrežje bo bolje uporabiti namensko napravo.

## 6. LITERATURA

- [1] (2009) Fizična plast. Dostopno na: [http://sl.wikipedia.org/wiki/Fizična\\_plast](http://sl.wikipedia.org/wiki/Fizična_plast)
- [2] (2009) Plast podatkovne povezave. Dostopno na: [http://sl.wikipedia.org/wiki/Plast\\_podatkovne\\_povezave](http://sl.wikipedia.org/wiki/Plast_podatkovne_povezave)
- [3] Andrew S. Tanenbaum, *Computer Networks Fourth Edition*, New Jersey: Upper Saddle River. 2003
- [4] (2009) Prenosna plast. Dostopno na: [http://sl.wikipedia.org/wiki/Prenosna\\_plast](http://sl.wikipedia.org/wiki/Prenosna_plast)
- [5] (2009) RFC793 - Transmission Control Protocol. Dostopno na: <http://www.faqs.org/rfcs/rfc793.html>
- [6] (2009) List of TCP and UDP port numbers. Dostopno na: [http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)
- [7] (2009) Poznavanje internetnih storitev. Dostopno na: <http://www.arnes.si/dokumenti/filtri/node5.html>
- [8] (2009) Linux. Dostopno na: <http://sl.wikipedia.org/wiki/Linux>
- [9] (2009) Povezavna in fizična plast © Mojca Ciglarič, 2009. Dostopno na: [http://ucilnica0809.fri.uni-lj.si/file.php/82/1\\_Fizicna\\_povezavna\\_plast.pdf](http://ucilnica0809.fri.uni-lj.si/file.php/82/1_Fizicna_povezavna_plast.pdf)
- [10] (2009) Iptables. Dostopno na: <http://en.wikipedia.org/wiki/Iptables>
- [11] (2009) Preverite povezavo v lokalnem omrežju. Dostopno na: <http://ndt.arnes.si/iperf.html>
- [12] (2009) Expect. Dostopno na: <http://en.wikipedia.org/wiki/Expect>
- [13] (2009) Gnuplot. Dostopno na: <http://en.wikipedia.org/wiki/Gnuplot>
- [14] (2009) Secure Shell. Dostopno na: [http://en.wikipedia.org/wiki/Secure\\_Shell](http://en.wikipedia.org/wiki/Secure_Shell)
- [15] (2009) Organizacija pisarniškega poslovanja. Dostopno na: <http://mms.fov.uni-mb.si/app/Ucbenik/multiplex.htm>
- [16] (2009) RFCs in HTML Format. Dostopno na: <http://rfc.activedomain.org/1000-1499/rfc1122.html>
- [17] (2009) RFC1323 - TCP Extensions for High Performance. Dostopno na: <http://www.faqs.org/rfcs/rfc1323.html>
- [18] (2009) RFC2018 - TCP Selective Acknowledgment Options. Dostopno na: <http://www.faqs.org/rfcs/rfc2018.html>

- [19] (2009) RFC2581 - TCP Congestion Control. Dostopno na: <http://www.faqs.org/rfcs/rfc2581.html>
- [20] (2009) RFC768 - User Datagram Protocol. Dostopno na: <http://www.faqs.org/rfcs/rfc768.html>
- [21] (2009) Registered port. Dostopne na: [http://en.wikipedia.org/wiki/Registered\\_port](http://en.wikipedia.org/wiki/Registered_port)
- [22] (2009) IPv4. Dostopno na: <http://ucilnica0809.fri.uni-lj.si/file.php/82/Vaje/IPv4.pdf>
- [23] (2009) TCP Offload Engine. Dostopno na: [http://en.wikipedia.org/wiki/TCP\\_Offload\\_Engine](http://en.wikipedia.org/wiki/TCP_Offload_Engine)
- [24] (2009) TCP in UDP. Dostopno na: <http://ucilnica0809.fri.uni-lj.si/file.php/82/Vaje/TCPinUDP-2.pdf>
- [25] (2009) Transportna plast © Mojca Ciglarič, 2009. Dostopno na: [http://ucilnica0809.fri.uni-lj.si/file.php/82/4\\_transportna\\_plast09.pdf](http://ucilnica0809.fri.uni-lj.si/file.php/82/4_transportna_plast09.pdf)
- [26] (2009) Aplikacijska plast © Mojca Ciglarič, 2009. Dostopno na: [http://ucilnica0809.fri.uni-lj.si/file.php/82/5\\_aplikacijska\\_plast.pdf](http://ucilnica0809.fri.uni-lj.si/file.php/82/5_aplikacijska_plast.pdf)
- [27] (2009) Ethernet. Dostopno na: <http://en.wikipedia.org/wiki/Ethernet>
- [28] (2009) IP Forwarding - How to make Windows XP as a Router. Dostopno na: <http://www.home-network-help.com/ip-forwarding.html>
- [29] (2009) RK Uvod © Mojca Ciglarič 2009. Dostopno na: [http://ucilnica0809.fri.uni-lj.si/file.php/82/0\\_uvod\\_rk.pdf](http://ucilnica0809.fri.uni-lj.si/file.php/82/0_uvod_rk.pdf)
- [30] (2009) ISO/OSI referenčni model. Dostopno na: [http://sl.wikipedia.org/wiki/ISO/OSI\\_referenčni\\_model](http://sl.wikipedia.org/wiki/ISO/OSI_referenčni_model)

## SEZNAM SLIK, TABEL IN IZPISOV

Slika 1.	Sestavni deli modela ISO OSI. [29] .....	4
Slika 2.	Sestavni deli modela TCP/IP. [29] .....	5
Slika 3.	Vsebina okvirja Ethernet. [9] .....	6
Tabela 1.	Seznam omrežij, njihove velikosti okvirjev (MTU) in teoretične hitrosti. ....	7
Slika 4.	Omrežna kartica (NIC) s priklopoma BNC in RJ45. [27] .....	8
Slika 5.	Usmerjevalnik in tri (pod)omrežja. [28] .....	10
Slika 6.	Sestavni deli datagrama IP. [22] .....	11
Slika 7.	Sestavni deli segmenta TCP. [22] .....	12
Tabela 2.	Seznam vrat, ki jih uporabljajo nekateri protokoli. ....	13
Slika 8.	Vzpostavljanje in rušenje povezave po protokolu TCP. [7] .....	14
Slika 9.	Sestavni deli segmenta UDP. [22] .....	15
Slika 10.	Diagram poteka – potovanja omrežnega paketa (paketa IP) po požarnem zidu v jedru Linux. ....	17
Slika 11.	Diagram primerov uporabe .....	19
Slika 12.	Diagram podatkovnih tokov. ....	20
Izpis 1.	Vsebina datoteke »testfile«. ....	21
Izpis 2.	Vsebina datoteke »settings«. ....	22
Slika 13.	Povezava merjencev in nadzornika, ter njihovi naslovi IP, maska omrežja in privzeti prehod. ....	23
Slika 14.	Diagram poteka optimalnega testiranja UDP. ....	25
Slika 15.	Diagram poteka optimalnega testiranja TCP .....	26
Slika 16.	Celotno testiranje 100 Mbit/s omrežja, med dvema merjencema, pri odvisnosti velikosti okna s protokolom TCP, 1500 B MTU in brez pravil v požarnem zidu. ....	28
Slika 17.	Celotno testiranje 100 Mbit/s omrežja, med dvema merjencema, pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in z nobenim pravilom v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete). ....	29
Slika 18.	Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri 1500 B MTU (korak 500). ....	30
Slika 19.	Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri 1500 B MTU (korak 500). ....	31
Slika 20.	Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri različnih velikostih MTU .....	32
Slika 21.	Optimalno testiranje 100 Mbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri različnih velikostih MTU. ....	33
Slika 22.	Celotno testiranje 100 Mbit/s omrežja pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in 5000 pravili v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete). ....	34
Slika 23.	Celotno testiranje 1 Gbit/s omrežja, med dvema merjencema, pri odvisnosti velikosti okna s protokolom TCP, 1500 B MTU in brez pravil v požarnem zidu. ....	35
Slika 24.	Celotno testiranje 1 Gbit/s omrežja, med dvema merjencema, pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in z nobenim pravilom v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete). ....	36
Slika 25.	Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri 1500 B MTU (korak 500). ....	37

Slika 26.	Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri 1500 B MTU (korak 500). ....	38
Slika 27.	Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom TCP pri različnih velikostih MTU.....	39
Slika 28.	Optimalno testiranje 1 Gbit/s omrežja pri odvisnosti števila pravil v požarnem zidu s protokolom UDP pri različnih velikostih MTU.....	40
Slika 29.	Celotno testiranje 1 Gbit/s omrežja pri odvisnosti hitrosti oddajanja s protokolom UDP, 1500 B MTU in 5000 pravili v požarnem zidu. Zelena prikazuje uspešen prenos (pakete), rdeča pa izgubljen prenos (pakete). ....	41